Theses and Dissertations   1. Thesis and Dissertation Collection, all items

1987

Control system simulation for a single-link flexible arm.

Park, Ki Soon.

http://hdl.handle.net/10945/22698

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

CONTROL SYSTEM SIMULATION
FOR A SINGLE-LINK FLEXIBLE ARM

by

Ki Soon Park

September 1987

Thesis Advisor   Liang-Wey Chang

T234317

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE				
4 PERFORMING ORGANIZATION REPORT NUMBER(S)		5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 69	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School		
6c ADDRESS (City State and ZIP Code) Monterey, California 93943-5000		7b ADDRESS (City State and ZIP Code) Monterey, Califonia 93943-5000		
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City State and ZIP Code)		10 SOURCE OF FUNDING NUMBERS		

		PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
CONTROL SYSTEM SIMULATION FOR A SINGLE_LINK FLEXIBLE ARM

12 PERSONAL AUTHOR(S)   Park, ki Soon

13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM _____ TO ____	14 DATE OF REPORT (Year Month Day) 87 sept.	15 PAGE COUNT 88

16 SUPPLEMENTARY NOTATION

17	COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Flexible Manipulator

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

To control a lightweight manipulator, a validated dynamic model based on the Equivalent Rigid Link System (ERLS), is used to formulate a control algorithm. The required torque to drive the manipulator to a desired angle is calculated. Since the control system includes an electrohydraulic actuator, the required current is also calculated by the inverse dynamics of the hydraulic servo. A computer simulation is performed with various loading conditions, gain values, and desired angles.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Chang, Code 69Ck	22b TELEPHONE (Include Area Code) 2632	22c OFFICE SYMBOL 69Ck	

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted

All other editions are obsolete

Control System Simulation
for a Single-Link Flexible Arm

by

Ki Soon Park
Major, Republic of Korean Air Force
B.S., Korean Air Force Accademy 1978
M.B.A., Chungnam University of Korea 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

ABSTRACT

To control a lightweight manipulator, a validated dynamic model based on the Equivalent Rigid Link System (ERLS), is used to formulate a control algorithm. The required torque to drive the manipulator to a desired angle is calculated. Since the control system includes an electrohydraulic actuator, the required current is also calculated by the inverse dynamics of the hydraulic servo. A computer simulation is performed with various loading conditions, gain values, and desired angles.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

8

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to my advisor, Professor Liang-Wey Chang, for his invaluable guidance and unselfish giving of his time in support of this research.

Appreciation is extended to the Korean Air Force Authority for the opportunity to study in Naval Postgraduate School.

I also want to thank my wife, Mi Sook and my daughter, Soo Yeun for their patience and encouragement in my study.

I. INTRODUCTION

A. MOTIVATION

The robotic manipulator plays an important role in a wide range of applications. Present and possible future applications focus on implementing automation in environments which are hostile to humankind. More demand is being placed on robots with higher speeds, smaller actuators, and higher payload capacity in applications.

In the past, efforts were made to control manipulators using rigid body assumptions. However, the demands for lightweight and higher performance manipulators have led to the consideration of flexible- body models. A new controller is needed for the flexible manipulators.

B. NATURE OF THE PROBLEM

Control of most manipulators is based upon dynamic models assuming perfectly rigid links. Accordingly, accurate manipulator motion control is dependent upon the great stiffness of the structural members. This assumption does not always hold, particularly at high-speed and heavy-load operations where a linkage may undergo severe elastic deformation due to the inertia. It can no longer perform its intended function in actual operations.

Lightweight flexible manipulators promise the advantages of increased speed of operation, ease of transportability, reduced material requirement, reduced power consumption, lowered mounting strength and rigidity requirement, and lower overall cost [Ref. 1]. On the other hand, the oscillatory behavior of the flexible links makes the end-point motions difficult to be controled. The central problem lies in the control of the end-point motion by applying the proper torques at the actuating point. Flexibility effect on manipulators is a necessary consideration in the move toward designing lighter, faster and more accurate robot systems. A dynamics model of manipulators with flexibility is needed for design and control purposes. To benefit from the advantages of lightweight, flexible manipulators, it is also necessary to implement a control design capable of achieving end effector positioning accuracy and stable control.

C.   LITERATURE REVIEW

The earliest studies concerning the control of flexible manipulators began in the early to mid 1970's. There has been considerable research recently in the development of flexible manipulator control strategies. A brief survey of this research follows.

Book [Ref. 2] introduced a modeling approach utilizing 4 × 4 transformation for control purposes. His model was limited in the assumption that the mass of the manipulator is negligible compared to the mass of the load. The assumption is acceptable in space applications where the manipulator is very light and operates at low speeds, but the assumption can not be applied to the industrial robotic manipulators in most cases.

Sunada and Dubowsky [Ref. 3] applied 4 × 4 transformation matrix and the Lagrangian approach to model the flexible system with geometrically irregular links by finite element techniques. Component Mode Synthesis was used to reduce the number of generalized coordinates and improve the computational efficiency without losing much accuracy. Based on this scheme, they can obtain the flexibility and mass properties of the links through available finite element programs. But, this model ignored the effect of the small motion interaction on the large motion. The consideration of the coupling effect between the small motion and the large motion was not completed.

Chang [Ref. 4] introduced the Equivalent Rigid Link System(ERLS) dynamic model of flexible manipulator. Global motions were separated into large motions and small motions. The ERLS described the kinematics of the large motion. The small motion displacements were described relative to the ERLS. The finite element method was used to discretize deformations and Lagrangian formation was used to derive the equations of motion. Two sets of coupled, non-linear ordinary differential equations - large motion equations and small motion equations resulted. The set of large motion equations were non-linear in both the large and small motion variables. The set of small motion equations were linear in the small motion variable but non-linear in the large motion variable. The model is complete being able to describe large motions, small motions and their coupling. The model presented the computational potential by using the Sequential Integration Method developed.

Petroka [Ref. 5] performed an experimental validation of the ERLS model. He designed a hydraulically actuated, single - link flexible arm. A cubic shape function was assumed to present the transverse displacement of the flexible manipulator. Tip position was determined from the motion picture studies.

Gannon [Ref. 6] served a continuation of the experimental validation of the ERLS model. A piezo-resistive accelerometer and strain gages were used for tip position and strain information. The natural mode shape functions of a beam was used to present the flexural motion of the single-link flexible arm in the Gannon's study. Results of the validation suggested the potential usefulness of the model in determining tip position.

Canon and Schmitz [Ref. 7] reported on experimental studies of the "End - Point Control of a Flexible One Link Robot". Their controller design was based upon a linearized and truncated model. Measurements available for control included rigid body angle and velocity, strain measurements and direct optical measurement of end-point position. Controller design was discussed from a variety of standpoints using both classical and modern control methods. The authors reported that the non -colocated(end -point position) feedback improves the speed of response at the cost of reducing the stability margins.

Flexibility effect is a very important consideration in designing lighter, faster and more accurate robot systems. In this research, the ERLS dynamic model is used to design a control system since the model presents a complete consideration of the system motion and exhibits computational superiority. Nonlinear motion is compensated in the control algorithm.

D.   RESEARCH OBJECTIVE

The objective of this research is to design a controller for single-link flexible manipulator. The validated dynamic model-the Equivalent Rigid Link System - is used to formulate the control algorithm. The control algorithm involves the computation of the required torque to drive the manipulator achieving a desired angle of the end of the flexible manipulator. Since the control system includes an electrohydraulic actuator, the required current is also calculated by inverse hydraulic dynamics. The ways to reduce the actuator effort and structural vibrations will be discussed.

II. MODEL FORMULATION FOR A SINGLE LINK FLEXIBLE MANIPULATOR

A. PHYSICAL PLANT

A flexible arm was designed by Petroka [Ref. 5] for the validation of the Equivalent Rigid Link System Model. The flexible arm shown in Figures 2.1 - 2.2 is a one meter long flexible structure which can be bend freely in the vertical plane but is stiff in the torsion and horizontal bending. The basic configuration of the arm includes two parallel, flexible steel, flat bars connected by the thin steel strips to transverse steel bridges. External loading is attached to the arm tip end transverse bridge by the securing of the load on the four welded studs shown in Figure 2.3. Table 1 shows the geometric and mass properties of the flexible arm. The flexible arm is driven by a electohydraulic system shown in Figure 2.1. The system consists of a York hydraulic power unit, a Berd- Johnson 3 - axis Hyd-Ro- Wrist, a Moog 760-100 servovalve, and associated Moog servocontroller and high pressure filter assembly [Ref. 5].

B. EQUIVALENT RIGID LINK SYSTEM FOR A SINGLE-LINK MANIPULATOR

The basic idea of ERLS [Ref. 4] is to separate the motion of the flexible link system into the large rigid-body motion and the small-motion displacement due to the flexibility of the structure. The large motion is defined by the movement of ERLS for a single - link flexible manipulator. The small-motion displacements are described with respect the ERLS. A schematic diagram of the ERLS for a single - link manipulator is shown in Figure 2.4.

The three generalized coordinates chosen are the large motion joint variable, θ, and two nodal displacements, $V(0)$ and $\varphi(0)$, which can be measured at the end of the link. The large motion joint variable, θ, is the angle between the ERLS and the horizontal axis of inertia coordinate. Homogenous transformations is applied for a deformed point on the single-link to obtain the absolute position of the point. The displacements for each point along the flexible arm are the functions of the location and time; therefore, it is necessary to discretize the deformations in terms of generalized coordinates in oder to apply Lagrangian's equation. The techniques of the Finite Element Method (FEM) are utilized for this discretization. The nodal displacements at the end of the single-link represent the small motion.

13

TABLE 1
ARM PARAMETERS

Parameter	Value	Unit
Arm Length (L)	0.9985	m
Beam Thickness(t)	0.003175	m
Beam Width(w)	0.0762	m
Distance Between Each Beam	0.05715	m
Arm Mass(m_t)	4.8565	kg
Modulus of Elasticity(E)	2.0 E11	N/m^2
Density/unit Volume(ρ)	7861.05	kg/m^3

After describing the kinematics relationships between the large and small motions,kinetics is introduced to complete the derivation of the equation motion.

The Lagrangian dynamics approach is used for the derivation due to its straightforward and systematic nature which is helpful in the analysis of complex systems. The kinetic energy is the sum of the kinetic energies of each link,actuator,and any loading. The potential energy is contributed the elastic strain energy and from gravity. Generalized forces are included due to any applied forces and damping forces. After making considerable effort in mathematical manipulations, rearrangements, and simplifications, the Lagrangian equations yield two sets of non-linear, coupled, second-order, ordinary differential equations. One set describes the large motions and the other set describes the small motions.

These equations are represented as follow;

$$M_{qq}\ddot{\theta} + M_{qn}\ddot{U} = F_q + TORQUE \tag{2.1}$$

14
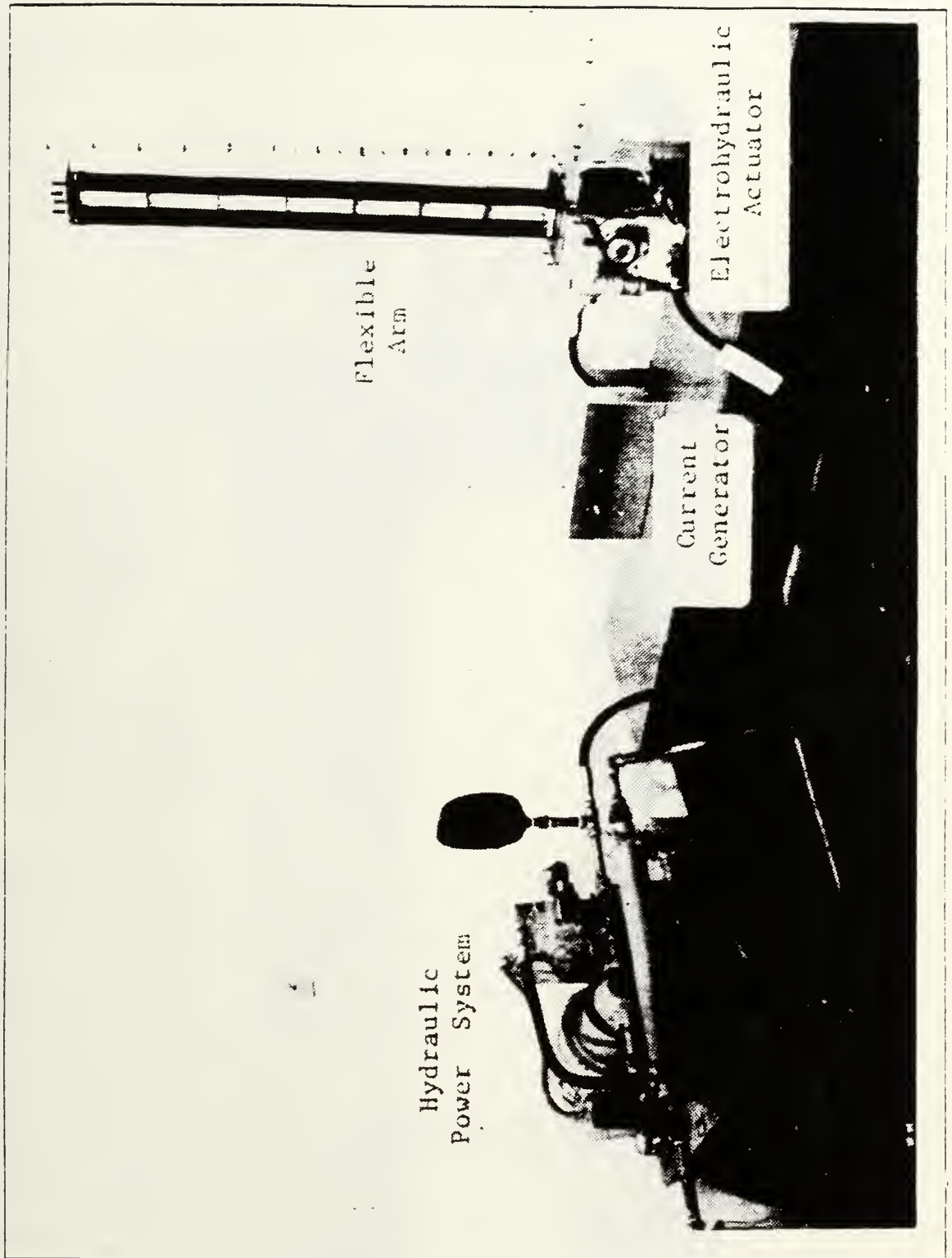
Figure 2.1   A Single-Link Flexible Manipulator System.

15
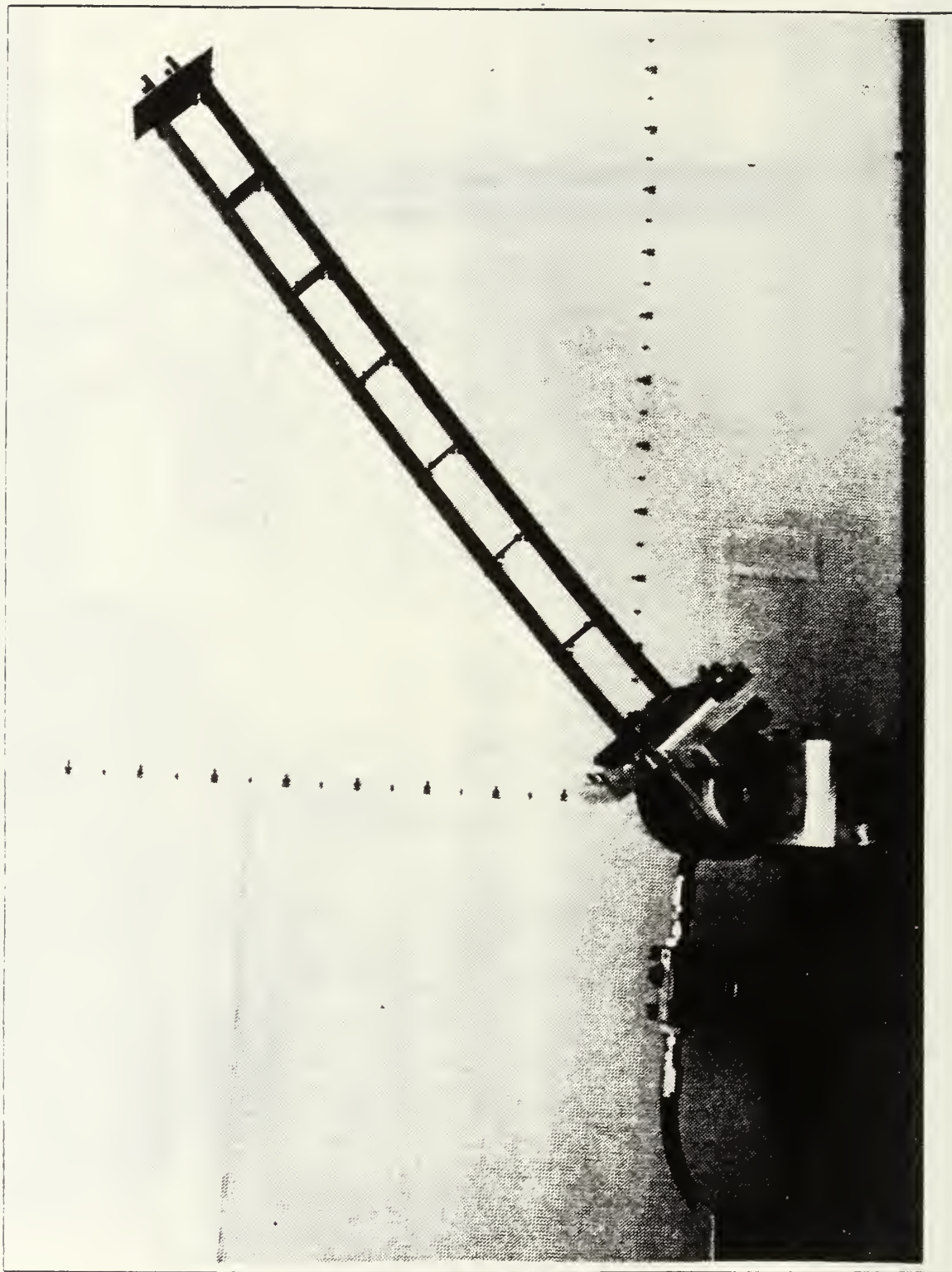
Figure 2.2   A Manipulator in Intermediate Position with 2.115kg Loading.
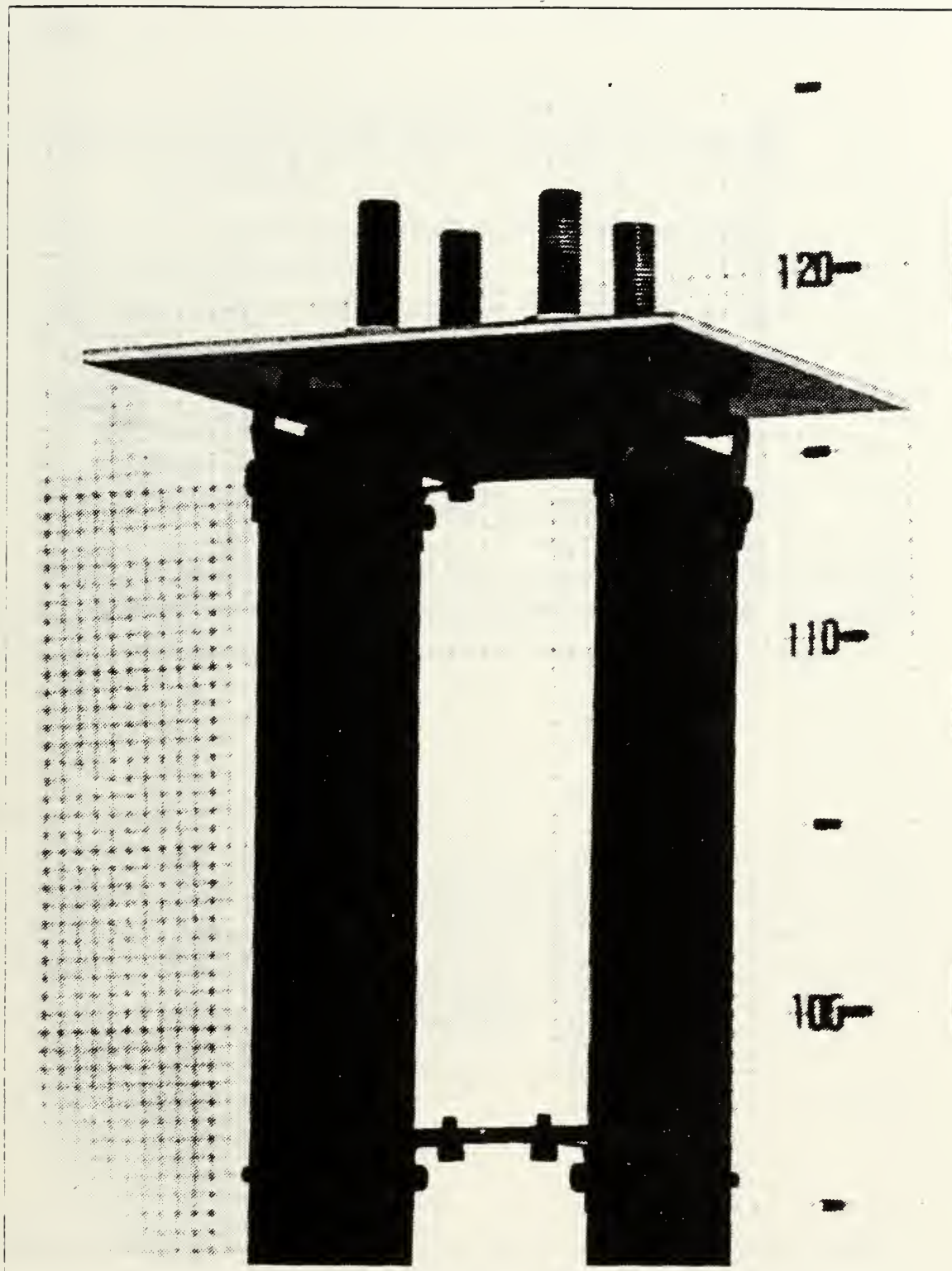
Figure 2.3   Details of Transverse Bridges and Tip Loading.

17
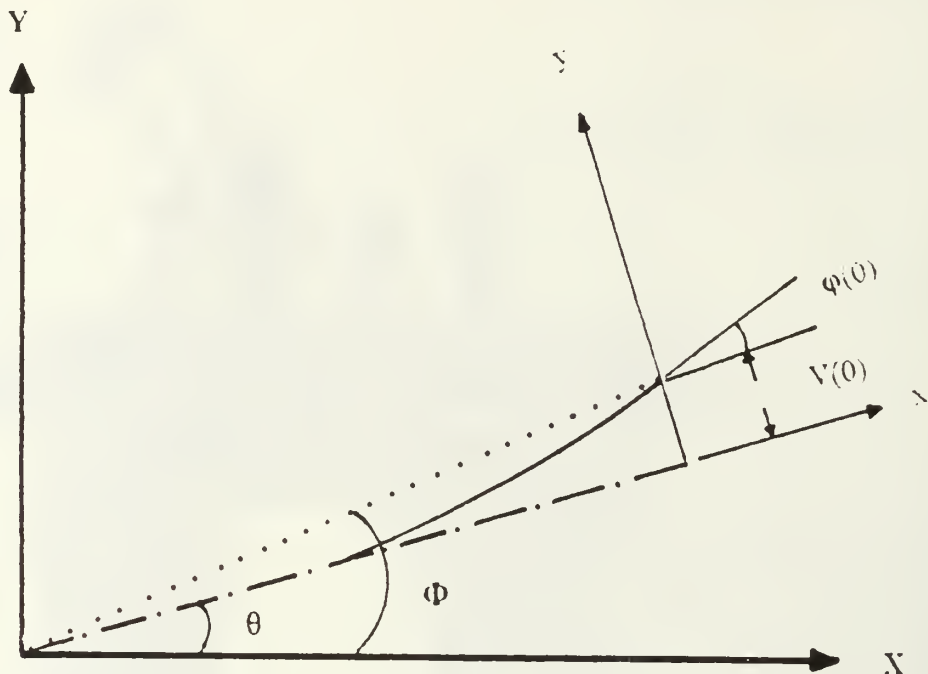
X, Y   : INERTIA COORDINATE

x, y   : LOCAL COORDINATE

θ : LARGE MOTION JOINT VARIABLE

$\varphi(0)$: ARM TIP SLOPE

$V(0)$: ARM TIP DEFLECTION

Φ : TOTAL ANGLE

Figure 2.4   The Equivalent Rigid Link System (ERLS).

$$M_{nq} \ddot{\theta} + M_{nn} \ddot{U} + K_n U = F_n \qquad (2.2)$$

where

M_{qq} = 1 × 1 effective inertia matrix for large motions

M_{qn} = 1 × 2 coupled inertia matrix of the small motion effect
on large motions

M_{nq} = 2 × 1 coupled inertia matrix of the large motion effect
on the small motions

M_{nn} = 2 × 2 effective inertia matrix for small motions

K_n = 2 × 2 stiffness matrix for small motions

F_q = 1 × 1 load vector for large motions

F_n = 2 × 1 load vector for small motions

θ = generalized coordinate of the large motions

U = 2 × 1 generalized coordinate of the small motions

The vector of small motion is limited to the transverse displacement of the end of the link. Axial deformation and torsion are neglected in this research for single-link manipulators. We assume that the displacement of the end of the link is small.

$$\mathrm{Tan}^{-1}(V/L) \sim V/L \qquad (2.3)$$

where V is a shorthand of V(0).

We also assume that the slope of the arm tip is not measured for the simple feedback control system. The tip displacement V(0) is the only generalized coordinate chosen for small motions. Then all coefficient of the matrixs of non-linear, coupled, second-order ordinary differential equations is reduced to 1 × 1 matrix. A total angle of the end of the link can be represented as follows;

$$\Phi = \theta + V/L \qquad (2.4)$$

$$\dot{\Phi} = \dot{\theta} + \dot{V}/L \qquad (2.5)$$

.

19

$$\ddot{\Phi} = \ddot{\theta} + \ddot{V}/L \qquad (2.6)$$

A more detailed derivation of the equations of motion is included in Appendix A.

C. SHAPE FUNCTION DERIVATION

The natural- mode shape function of a beam is needed to present the flexural motion of the single-link flexible arm [Ref. 4]. The flexible manipulator arm is modeled as a continuous Euler-Bernoulli cantilever beam. The transverse displacement for the single link flexible arm is represented in the following forms;

$$
\begin{aligned}
U\{x,t\} &= a_1 \{t\} \chi_1 \{x\} + a_2 \{t\} \chi_2 \{x\} \\
&= a_1 \{A_1{}' \{\cos\beta_1 x + \cosh\beta_1 x\} + \{\sin\beta_1 x + \sinh\beta_1 x\}\} \\
&\quad + a_2 \{A_2{}' \{\cos\beta_2 x + \cosh\beta_2 x\} + \{\sin\beta_2 x + \sinh\beta_2 x\}\}
\end{aligned} \qquad (2.7)
$$

where
$$\beta_1 L = 1.875104069$$
$$\beta_2 L = 4.694091133$$

$$A_i{}' = \frac{\sin\beta_i L + \sinh\beta_i L}{\cos\beta_i L + \cosh\beta_i L} \qquad (2.8)$$

Upon reduction, Gannon [Ref. 6] yielded a 3 × 2 shape function matrix Ψ, and the 2 × 1 nodal displacement vector, U. a 3 × 1 deformation vector, d , was expressed as follow;

$$
d = \Psi U = \begin{bmatrix} 0 \\ 0 \\ V(x) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ M & N \end{bmatrix} \begin{bmatrix} V(0) \\ \varphi(0) \end{bmatrix}
$$

The coefficents M and N are defined as;

$$
\begin{aligned}
M &= \{C_1 \{A_1{}' \{ \cos\beta_1 x + \cosh\beta_1 x \} + \{ \sin\beta_1 x + \sinh\beta_1 x\}\} \\
&\quad + C_2 \{ A_2{}'\{\cos\beta_2 x + \cosh\beta_2 x\} + \{\sin\beta_2 x + \sinh\beta_2 x\}\}\} \\
N &= \{C_3 \{A_1{}' \{ \cos\beta_1 x + \cosh\beta_1 x \} + \{ \sin\beta_1 x + \sinh\beta_1 x\}\}
\end{aligned}
$$

20

$$+ C_4 \{ A_2'\{\cos\beta_2 x + \cosh\beta_2 x\} + \{\sin\beta_2 \ x + \sinh\beta_2 x\}\}\}$$

where

$$C_1 = 2\beta_2 / \{4 A_1' \beta_2 - 4 A_2' \beta_1\}$$
$$C_2 = -2\beta_1 / \{4 A_1' \beta_2 - 4 A_2' \beta_1\}$$
$$C_3 = -2 A_2' / \{4 A_1' \beta_2 - 4 A_2' \beta_1\}$$
$$C_4 = 2 A_1' / \{4 A_1' \beta_2 - 4 A_2' \beta_1\}$$

In this research, since the slope function of the arm tip is not measured, the shape function, Ψ, is reduced to a 3×1 matrix. Now the 3×1 deformation vector can be rewritten as follows;

$$d = \Psi U = \begin{bmatrix} 0 \\ 0 \\ V(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ M \end{bmatrix} \begin{bmatrix} V(0) \end{bmatrix}$$

New shape function matrix is now in a convenient form for computer coding. A more detailed derivation of the shape function was included in Gannon's work.

D.   HYDRAULIC ACTUATION

The single-link flexible manipulator uses electrohydraulic actuation to drive the plant. The inclusion of the electrohydraulic power system involves the transformation of an input current to an output torque. Figure 2.5 illustrates the relationship of hydraulic dynamics to the overall system.

The hydraulic dynamics is represented by servovalve dynamics and actuator dynamics. Moog, the manufacture of the servovalve, simplified the description of servovalve dynamics to a single equation [Ref. 8].

$$Q = I K \sqrt{P_v} \tag{2.9}$$

where

Q = flow delivered from servovalve

I = input current

K = valve sizing constant,contributes to hydraulic system damping

P_v = valve pressure drop,P_s-P_L

21
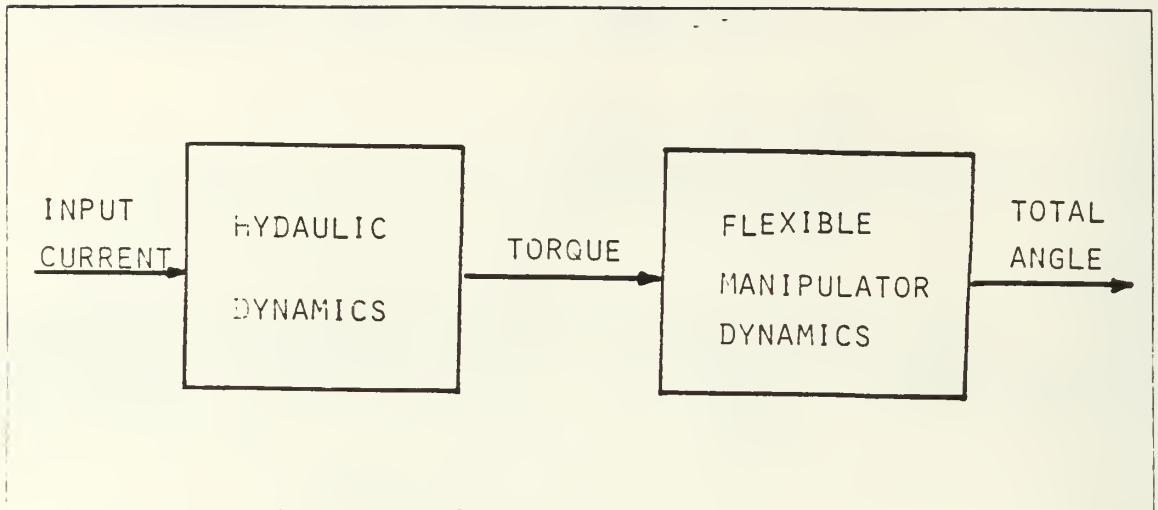
Figure 2.5   An overall Plant.

Actuator dynamics consists of a form of the continuity equation and the torque output equation. Actuator dynamics consists of a form of the continuity equation and the torque output equation [Ref. 9].

$$Q = D_m \dot{\theta} + C_{tm} P_L + \frac{V_t \dot{P}_L}{4 \beta_e} \tag{2.10}$$

$$T_d = \eta_t P_L D_m \tag{2.11}$$

Where

Q = flow delivered to actuator

$D_m \dot{\theta}$ = flow causing actuator rotation

β_e = effective bulk modulus of fluid

V_t = total compressed volume including actuator lines and chambers

$V_t \dot{P}_L / (4 \beta_e)$ = compressability flow

T_d = torque required to overcome inertia and move the load

η_t = torque efficiency

P_L = load pressure drop

D_m = motor displacement

$C_{tm} P_L$ = leaking flow in actuator

The preceding hydraulic dynamic equations and relationships are incorporated into the main computer program for solving the dynamic equation of motion.

III. CONTROL SCHEME

A.   REQUIRED TORQUE

In feedback control system, desired angles are compared with the measurements of the actual system responses. The differance is used via the chosen control algorithm. The control algorithm calculates the required torque to derive the manipulator achieving a desired angle of the end of the arm.

Recall the equations that can be written as two sets of equations - large motions and small motions equations - and the total angle equation of the end of the arm for derivation of the required torque equation.

$$M_{qq} \ddot{\theta} + M_{qn} \ddot{V} = F_q + TORQUE \tag{3.1}$$

$$M_{nq} \ddot{\theta} + M_{nn} \ddot{V} + K_n V = F_n \tag{3.2}$$

where

M_{qq} $= 1 \times 1$ effective inertia matrix for large motions

M_{qn} $= 1 \times 1$ coupled inertia matrix of the small motion effect on large motions

M_{nq} $= 1 \times 1$ coupled inertia matrix of the large motion effect on the small motions

M_{nn} $= 1 \times 1$ effective inertia matrix for small motions

K_n $= 1 \times 1$ stiffness matrix for small motions

F_q $= 1 \times 1$ load vector for large motions

F_n $= 1 \times 1$ load vector for small motions

θ $=$ generalized coordinate of the large motions

v $= 1 \times 1$ generalized coordinate of the displacments

$$\theta = \Phi - V/L \tag{3.3}$$

24

$$\dot{\theta} = \dot{\Phi} - \dot{V}/L \tag{3.4}$$

$$\ddot{\theta} = \ddot{\Phi} - \ddot{V}/L \tag{3.5}$$

Equations 3.3-3.5 can be substituted into Equations 3.1 and 3.2 and rearranged as follows;

$$M_{qq}(\ddot{\Phi} - \ddot{V}/L) + M_{qn} \ddot{V} - F_q = \text{TORQUE} \tag{3.6}$$

$$M_{nq}(\ddot{\Phi} - \ddot{V}/L) + M_{nn} \ddot{V} + K_n V - F_n = 0 \tag{3.7}$$

After eliminating V, a governing equation for Φ is given as;

$$(M_{qq} - D M_{nq}) \ddot{\Phi} + (D F_n - D K_n - F_q) V = \text{TORQUE} \tag{3.8}$$

where

$$D = \frac{(M_{qn} - (M_{qq}/L))}{(M_{nn} - (M_{nq}/L))} \tag{3.9}$$

Let
$$N = M_{qq} - D M_{nq}$$
Equation 3.8 can be expressed in short as follows;

$$N \ddot{\Phi} + FC (V, \dot{V}, \theta, \dot{\theta}) = \text{TORQUE} \tag{3.10}$$

where

$$FC(V, \dot{V}, \theta, \dot{\theta}) = (D F_n - D K_n - F_q) V$$

Φ is measured and fed back to controller. The required torque to drive the manipulator achieving a desired angle of the end of the arm is calculated via a controller. The inverse problem is applied to calculate the required torque. Figure 3-1 shows a schematic diagram of the feedback control system.

The equation of the required torque is represented as follows;

$$T_r = N \, (\, \ddot{\Phi}_d + K_v \, (\dot{\Phi}_d - \dot{\Phi}) + K_p \, (\Phi_d - \Phi)) + FC \tag{3.11}$$

where

T_r = a required torque
$\ddot{\Phi}_d$ = a desired angular acceleration
$\ddot{\Phi}$ = a total angular acceleration
$\dot{\Phi}_d$ = a desired angular velocity
$\dot{\Phi}$ = a total angular velocity
Φ_d = a desired angle
Φ = a total angle
K_v = a velocity gain
K_p = a position gain

The torque of the Equation 3.10 is equals to the required torque of the Equation 3.11. System errors of a second order control system can be represented as follows;

$$(\, \ddot{\Phi}_d - \ddot{\Phi} \,) + K_v \, (\dot{\Phi}_d - \dot{\Phi}) + K_p \, (\Phi_d - \Phi) = 0 \tag{3.12}$$

B.   REQUIRED CURRENT

Since the control system includes an electrohydraulic actuator, the inverse hydraulic dynamic involves the transformation of input torque to an output current. For the input current to the hydraulic dynamics, the inverse hydraulic dynamics can be applied to calculate the required current according to the required torque. A desired load pressure drop and desired flow should be calculated by the inverse hydraulic dynamic equations.

$$P_{Lr} = T_d \, / (\eta_t \, D_m) \tag{3.13}$$

26
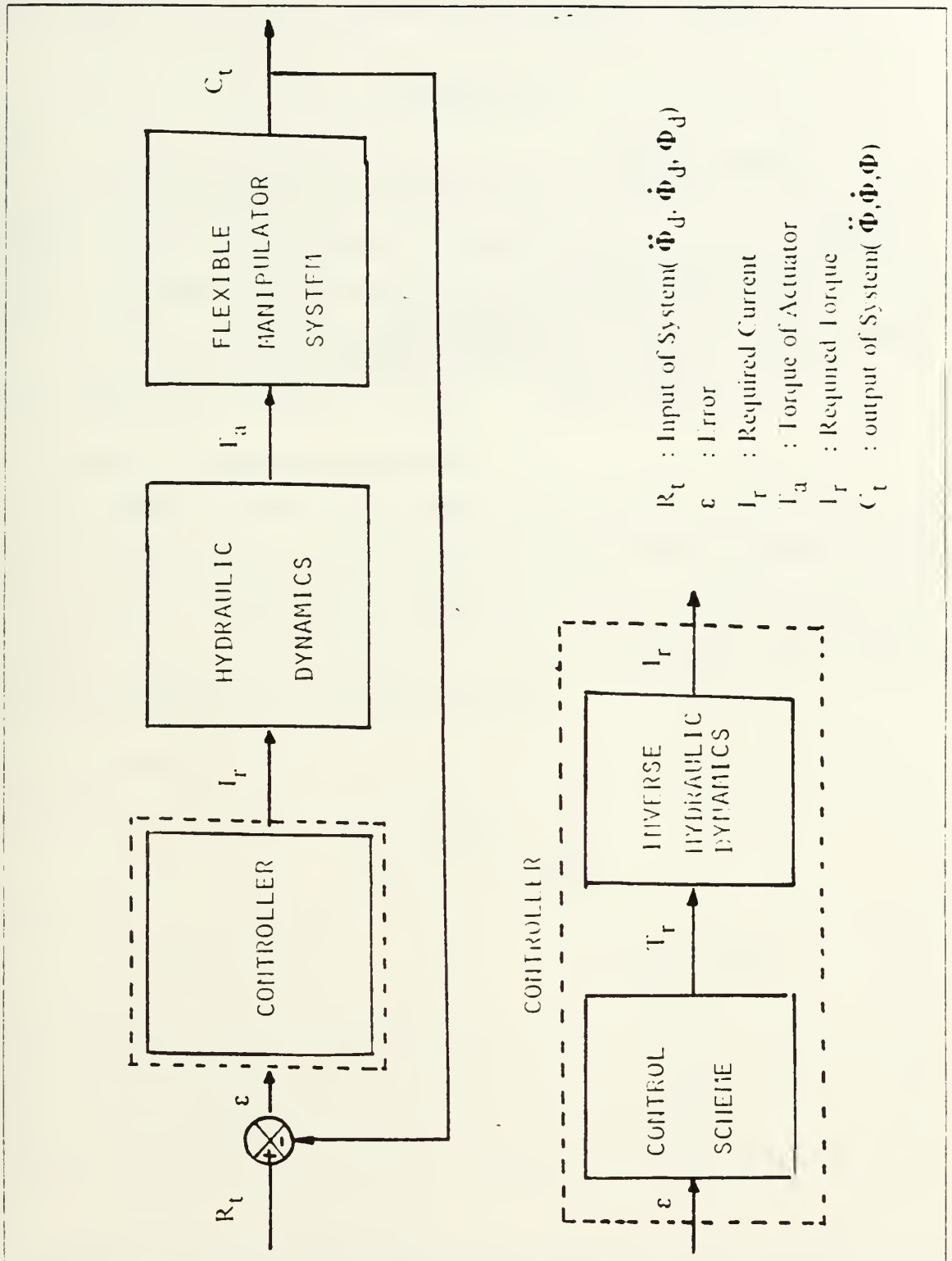
Figure 3.1   A Schematic Diagram of the Feedback Control System.

27

$$Q_r = D_m \dot{\theta} + C_{tm} P_{Lr} + (V_t \dot{P}_{Lr})/(4 B_e) \quad \text{-} \text{-} \tag{3.14}$$

$$I_r = Q_d /(K \sqrt{P_v}) \tag{3.15}$$

where

T_r = a required torque

P_{Lr} = a required load pressure drop

Q_r = a required flow delivered to actuator

I_r = a required current

IV. RESULT

A control algorithm for a single-link flexible manipulator is based upon the Equivalent Rigid Link System Model. The control algorithm calculates the required torque to control the end position of the manipulator. Comparisons of the required torque and required current are made in terms of the three parameters (i.e., loading condition, gain value, and desired angle). Additionally, the way to reduce the required currents and required torques is discussed. The effect of the current saturation to system dynamics behavior is also presented.

A.   INPUTS AND MOTION OUTPUTS

The loading conditions are no load, a 2.115 kg ,and a 4.233kg load. The external loading is attached to the arm tip end transverse bridge by securing the load on the four welded studs. The initial condition for all runs is the horizontal position of the flexible arm. The initial tip deformations as determined by the Euler-Beroulli cantilever beam theory are converted to an angle. These angles in Table 2 are the initial conditions input into the simulation program.

TABLE 2 INITIAL CONDITIONS	
Load (Kg)	Angle (radian)
0.0	-0.06111
2.115	-0.14628
4.233	-0.23155

Only point to point control in open space is considered. An input variable to the closed-loop system is the desired angle of the end-point. The desired angular acceleration and the desired angular velocity are zero in this research. To compare the

system responses, two step inputs, which are two desired angles of the end-point,(i.e., 1 radian and 1.5 radian), are applied respectively.

The outputs of the system responses for the step input of the desired angle are total-angle accelerations, total-angle velocities and total angles of the end-point, which are also fed back to the controller. First, the desired angle of 1 radian is input as the step function. Figures 4.1-4.3 show the movements of the end-point for all loading conditions. Each figure has three curves according to selected gain values for all loading conditions. In this research, a velocity gain (K_v) and a position gain (K_p) are selected for critically damped systems. In other words, the movements of the total angle of the end-point will not exceed the desired angle of the end-point. The position gain can be expressed in terms of the velocity gain for the critically damped systems as fellows;

$$K_p = K_v^2 / 4$$

Table 3 shows the three cases of gain values for this research:

TABLE 3 GAIN VALUES		
Case	Velocity Gain	Position Gain
1	2	1
2	4	4
3	8	16

For the evaluation of the system performances for the step input, a settling time corresponding to a 2 % tolerance is measured in terms of time constant($T = 1 / K_p^{1/2}$). Settling time($t_s = 4 T$) are 4 sec, 2 sec and 1 sec according to the position gains 1, 4 and 16. All results of the total angles of the end-point meet error criteria of 2 % according to the gain values. The figures of the total angle of the end-point with larger gain values show faster movement. Another observation is that the total angle

of the end-point with the same gain value have the same movement in terms of the time constant for the different loading conditions. These results indicate that the nonlinear plant is controlled to perform a linear behavior. Errors of the system can be represented as the second-order control system as follows;

$$(\ddot{\Phi}_d - \ddot{\Phi}) + K_v (\dot{\Phi}_d - \dot{\Phi}) + K_p (\Phi_d - \Phi) = 0$$

Figures 4.4-4.5 show the total-angle velocity and total-angle acceleration approaching the desired angle of 1 radian for the 4.233 kg loading condition. All initial velocities start from zero and being increasing to achieve the desired angle with fast movement as gain values are increasing during transients and converge to zero velocity as the total angle of the end-point approachs to the desired angle.

Figure 4.6 shows the comparison of the total angle response for the step inputs of 1 radian and 1.5 radian with the position gain of 4, the velocity gain of 4 and 2.115 kg loading condition. The total angle movement of the 1.5 radian is faster than the movement of the desired angle of 1 radian. But both of the total angle response have same settling time since they have the same position (4) and velocity gain values(4). Figures 4.7-4.8 show the comparisons of the total-angle velocity and total-angle acceleration between the 1 radian and 1.5 radian of the desired angle with same position gain for the 2.115kg loading condition. The figures of the total-angle velocity and total-angle acceleration indicate that total-angle velocities and accelerations are increasing as desired angle is increasing with same gain value during the transients. The maximum of total-angle velocities and total-angle accelerations for the desired angles are shown in Table 4.

B.   REQUIRED TORQUES

In the feedback control system, the desired angle of the end-point is compared with the measurements of the actual system responses. The differences are used via the chosen control algorithm to calculate the required torque. In other words, the control algorithm calculates the required torques to drive the manipulator to achieve the desired angle of the end-point according to the chosen gain values. The equation of the required torque was represented as follows;

$$T_r = N (\ddot{\Phi}_d + K_v (\dot{\Phi}_d - \dot{\Phi}) + K_p (\Phi_d - \Phi)) + FC$$

31

TABLE 4
MAXIMUM OF TOTAL-ANGLE VELOCITY AND TOTAL-ANGLE ACCELERATION

Desired Angle(rad)	Maximum Angular Velocity(rad/sec)	Maximum angular Acceleration(rad/sec^2)
1.0	0.974	37.61
1.5	1.29	37.78

Figures 4.9-4.11 show the required torques to drive the manipulator in order to achieve the desired angle of 1 radian for all loading conditions. The desired angle of 1 radian is input as a step function. All figures have three curves according to the three gains in Table 1. Table 5 shows the maximum of the required torques according to the gain values and the loading condition. The figures for the required torques and Table 5 show expected results; the required torques during the transient period are increasing to deliver the load and to overcome the inertia moment as gain values are increasing, and rapidly settle to the steady-state value. These required torques are related to the movements of total angle of the end-point. For the faster movements of the total angles, more required torque should be applied to the physical plant. Because of no steady-state error, the required torques at the steady-state (the total angle reaches to the desired angle) have same values for the same desired angle even though the gains are different.

Figure 4.12 shows the required torques for desired angles(i.e.,1 radian and 1.5 radian). The comparison for the desired angles indicates that the maximum of the required torque depend upon the movements of the total angle is increasing as the desired angle is increasing.

C.   CURRENT CALCULATIONS

The control algorithm includes an electrohydraulic actuator. The input current to the manipulator can be calculated from the inverse electrohydraulic dynamics according to the required torque. Figures 4.13-4.15 represent the graphical results of

Load(kg)	Postion Gain	Torque(N-m)
	TABLE 5 MAXIMUM REQUIRED TORQUE	
Load(kg)	Postion Gain	Torque(N-m)
No Load	1	82.559
	4	84.041
	16	89.969
2.115	1	148.11
	4	155.64
	16	185.76
4.233	1	125.73
	4	139.68
	16	183.44

the required currents for all loading conditions. Each figure has the three curves according to three gain values. Figures indicate that the required currents have fluctuations based upon required torques during the transients, and settle to the steady- state values.

D. THE REDUCTIONS OF THE REQUIRED TORQUE

The magnitudes of the required torques and required currents are related to the selection of size of the electrohydraulic actuator. The selections of the electrohydraulic actuator which depend upon the required torques and required currents, are also related to the weight,and cost. To reduce the required torques and required currents is very important in terms of the selection of the size of electrohydraulic actuator. In this research, a planned desired angle trajectory is used as the input replacing the step input.

The planned input is the combination of ramp and step function applied to the 4.233kg load with the position gain of 16. The initial point of the ramp function coincides with the initial angle of the flexible arm. The slope of the ramp function is selected as 2. Figure 4.16 shows the planned input of the desired angle as the combination function. Figures 4.17-4.19 show comparisons of the results between the step input and planned input. The difference of the maximum required torque for input functions is tabulated in Table 6.

<table>
<tr><td colspan="2" align="center">TABLE 6
MAXIMUM TORQUE FOR INPUT FUNCTIONS</td></tr>
<tr><td align="center">Input Function</td><td align="center">Max. Torque(N-m)</td></tr>
<tr><td align="center">Step input</td><td align="center">183.41</td></tr>
<tr><td align="center">Planned Input</td><td align="center">106.58</td></tr>
</table>

Table 6 indicates that the plan of the input function for the desired angle could results in the reduction of the maximum required torque and maximum required current. The total angle responses might be slower. We can expect different reductions of the required torques or required currents for same loading conditions and gain value when we change the input function of the desired angle.

E.   THE REDUCTION OF VIBRATION

The system damping is considered to reduce the structural vibration. A damping ratio is applied to the small motion equation as follows;

$$M_{nq} \ddot{\theta} + M_{nn} \ddot{V} + 2 \zeta W_n \dot{V} + K_n V = F_n$$

The no load fundamental frequency is predicted to be 2 HZ. A damping ratio(ζ) of 0.5 is applied to no load condition with the position gain of 2. Figures 4.20-4.23 show graphical results of the comparison of the no damping case with the damping case that has a 0.5 damping ratio. These result show that the movement of the end-point of the damping case is faster than the movement of the no damping. The system damping

yields the noticeable reduction of the vibration of small motion. The required torques of the system damping case show less fluctuations.

F.   THE ELECTROHYDRAULIC SATURATION

The previous results for the required torques and required currents are for an ideal condition. When the electrohydraulic actuator is saturated at 15 milliamp current, all current exceeded 15 milliamp will be limited at 15 milliamp current. Figures 4.24-4.26 show the effect of the saturation for the 4.233 kg loading condition with the position gain of 16. The total angle responses with the saturated current show slightly slower movements compared to the ideal movement during transients. The required torques are also reduced due to the saturated current during the initial transients.
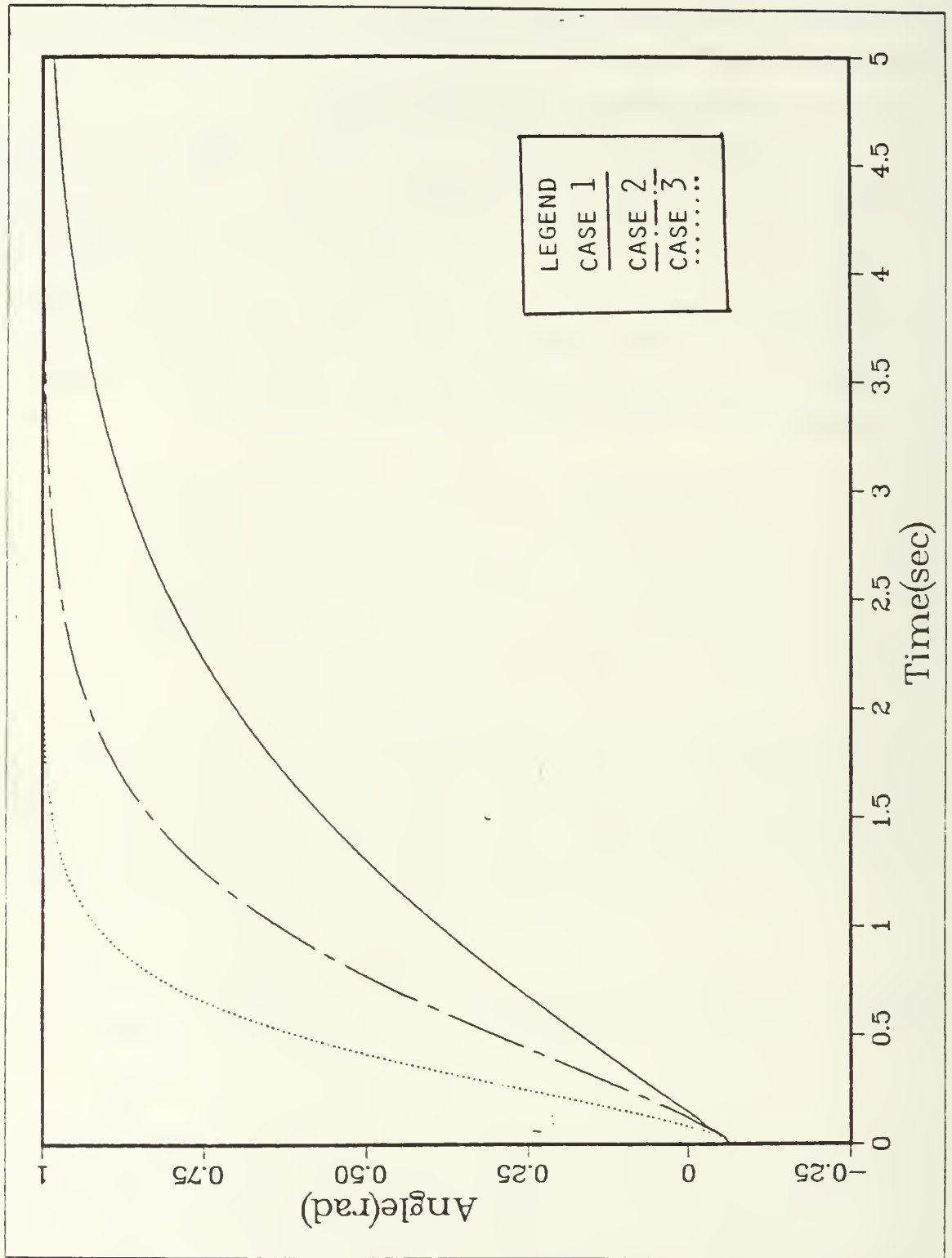
Figure 4.1   Total Angle Response   (0 kg).

36
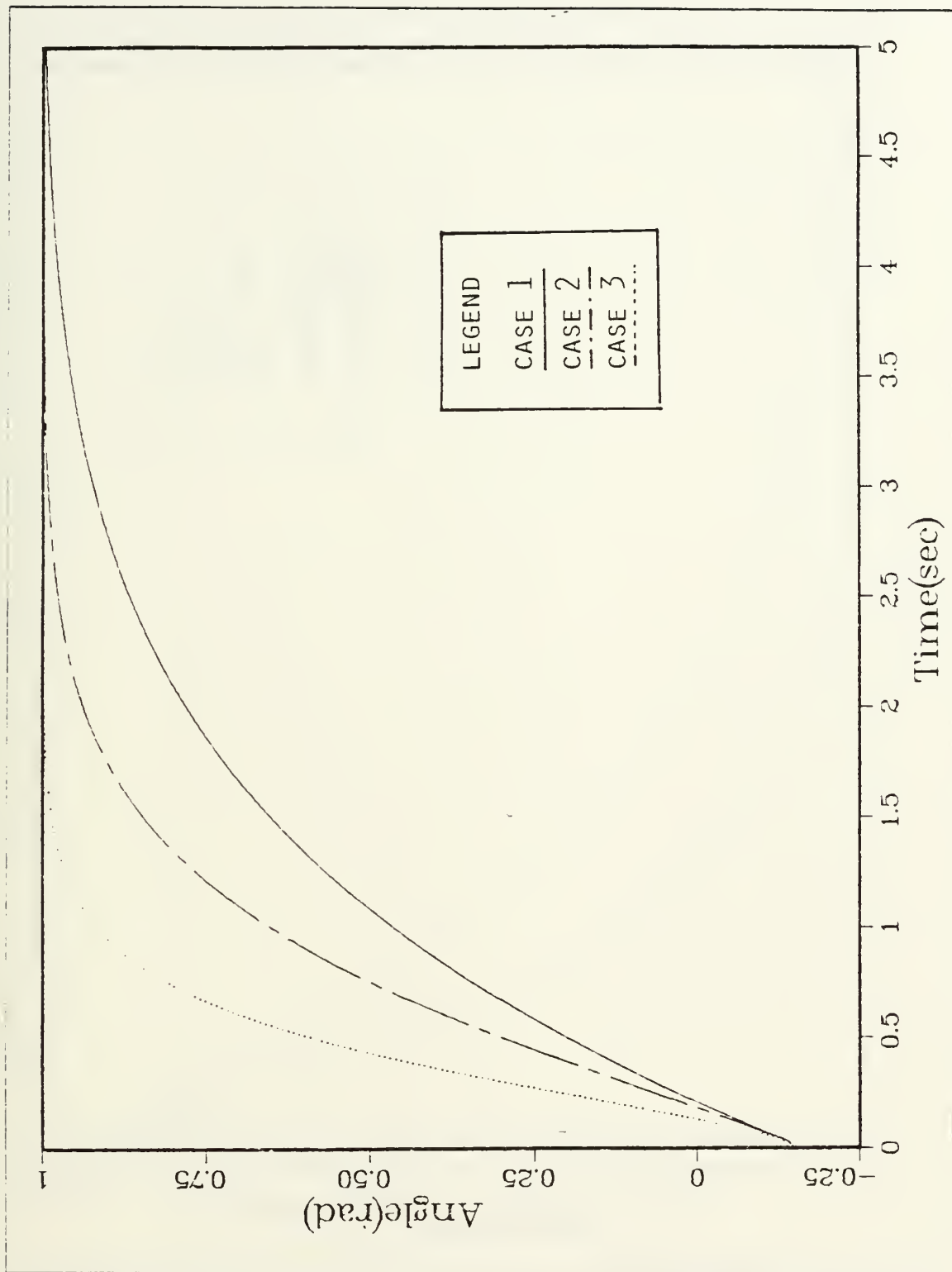
Figure 4.2   Total Angle Response   (2.115 kg).
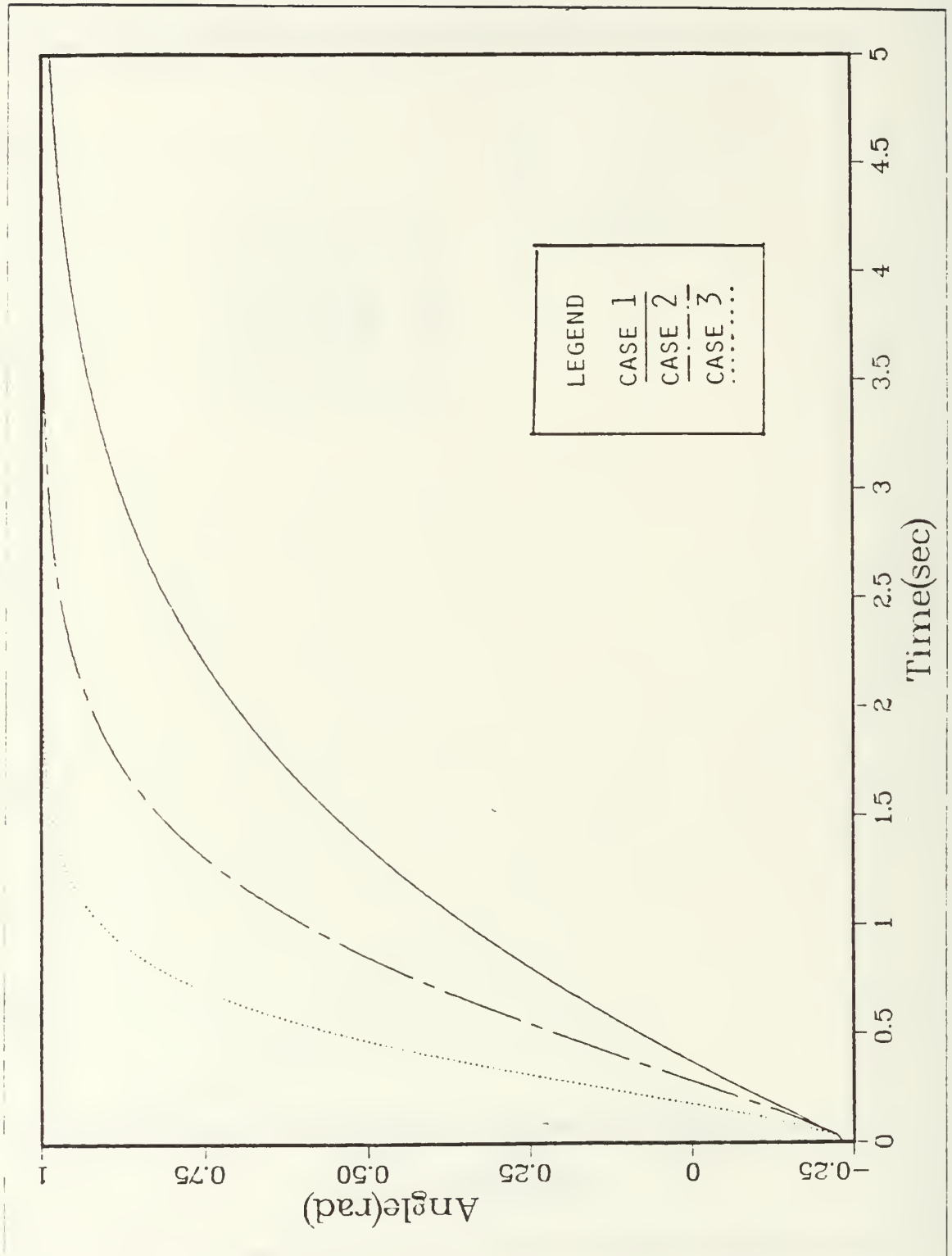
Figure 4.3   Total Angle Response (4.233 kg).

38
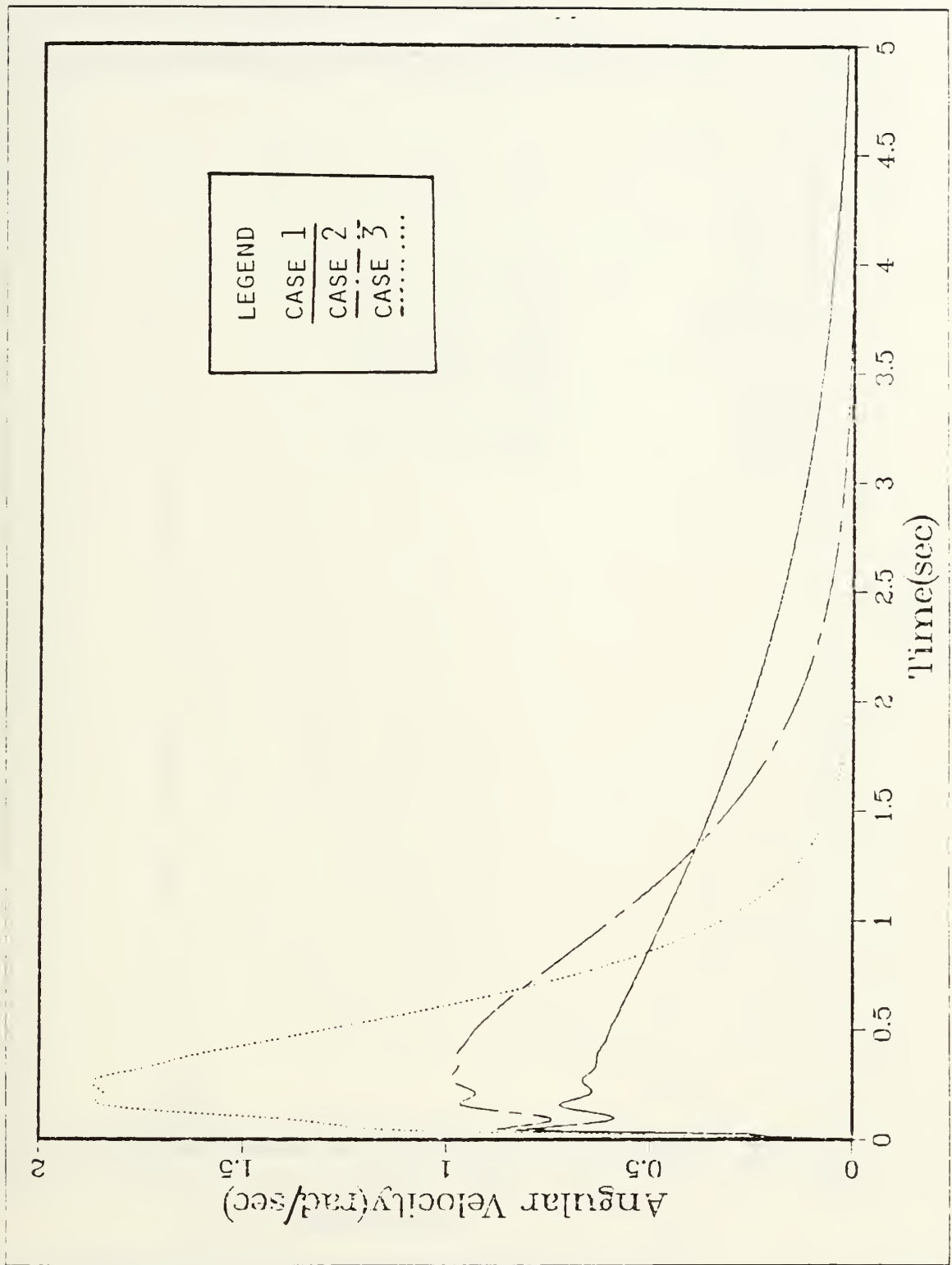
Figure 4.4 Total-Angle Velocity (4.233kg).

39
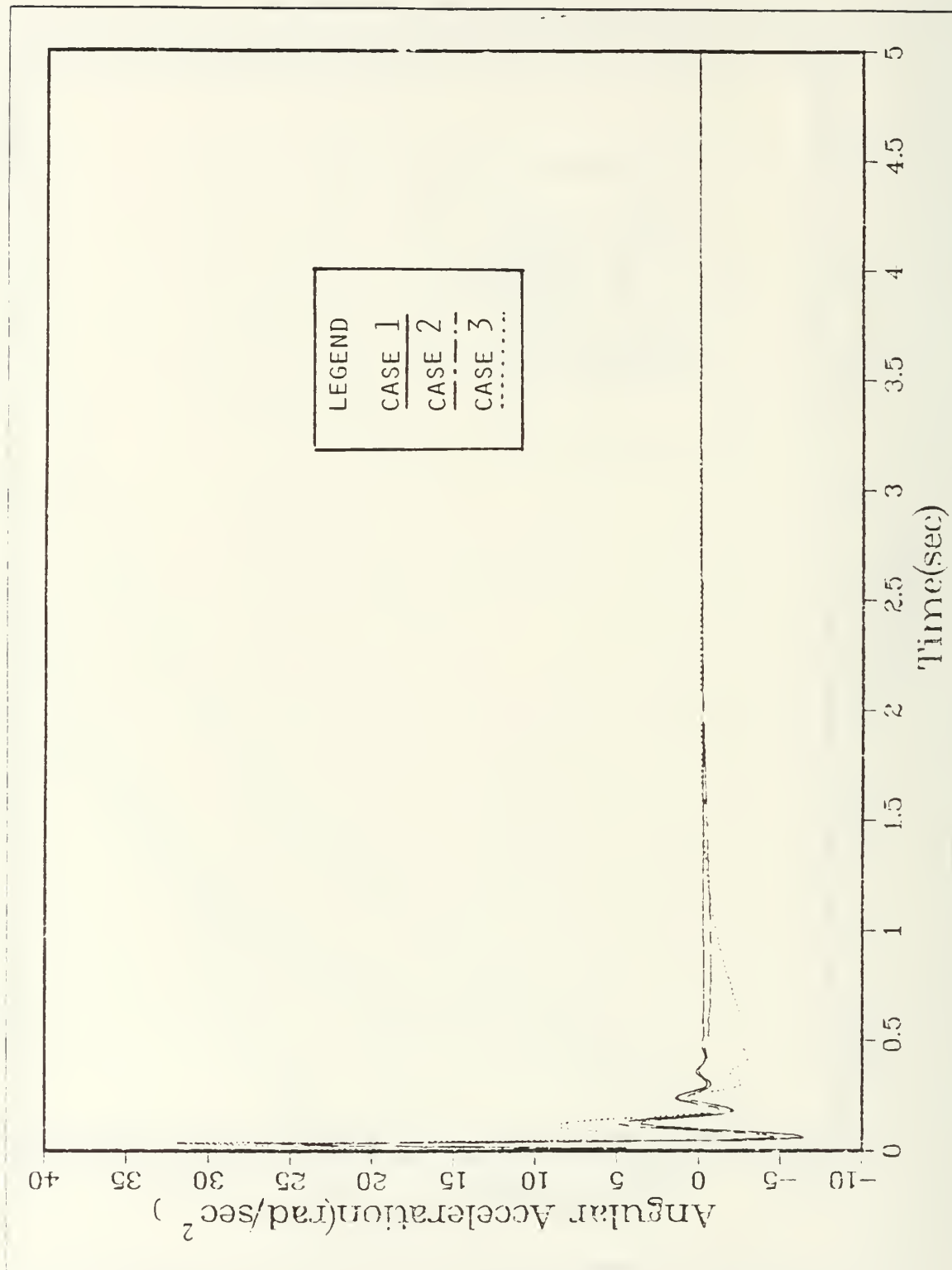
Figure 4.5   Total-Angle Acceleration(4.233kg).

40
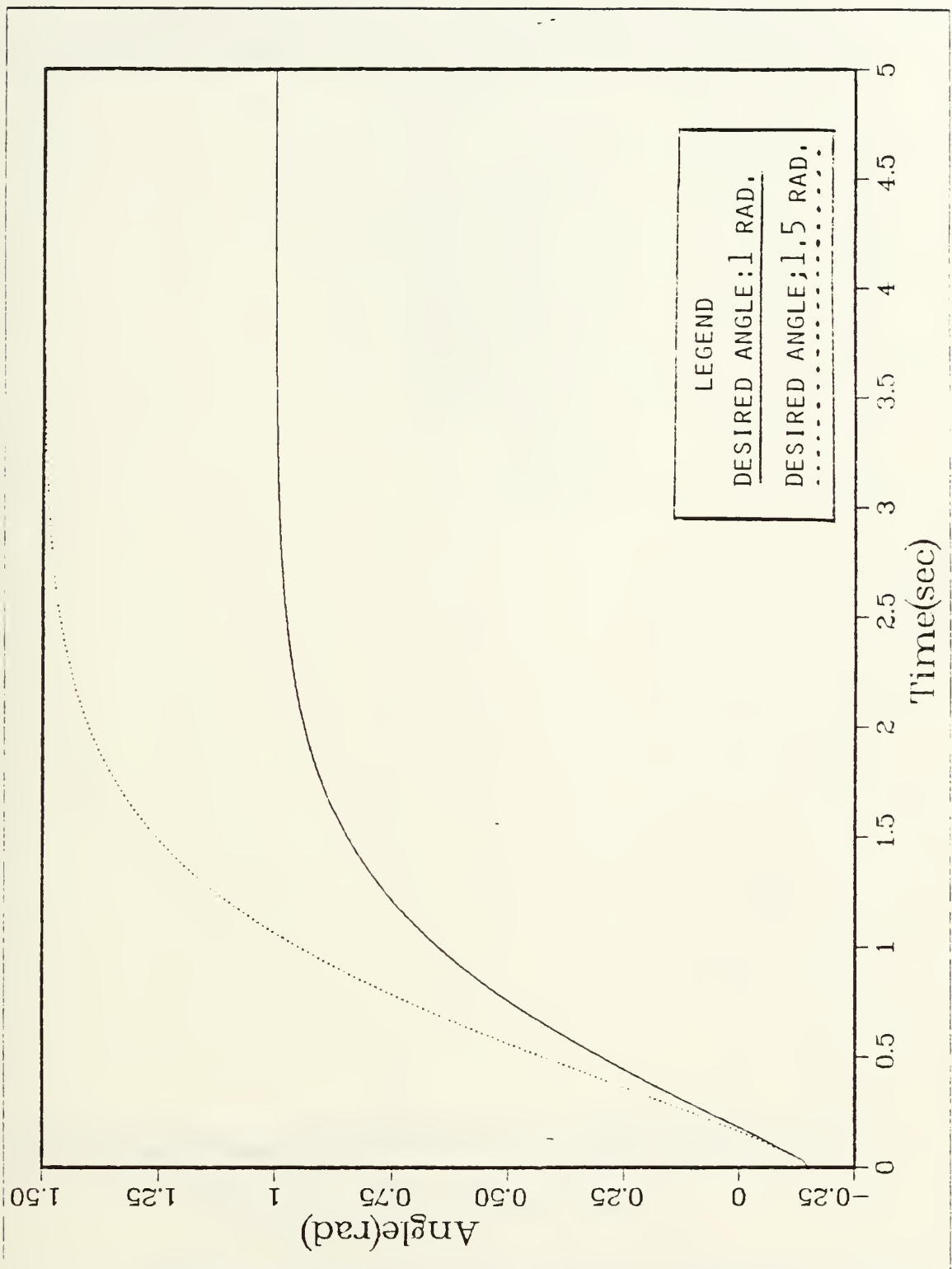
Figure 4.6 Total Angle Response for Various Desired Angles(2.115kg).

41
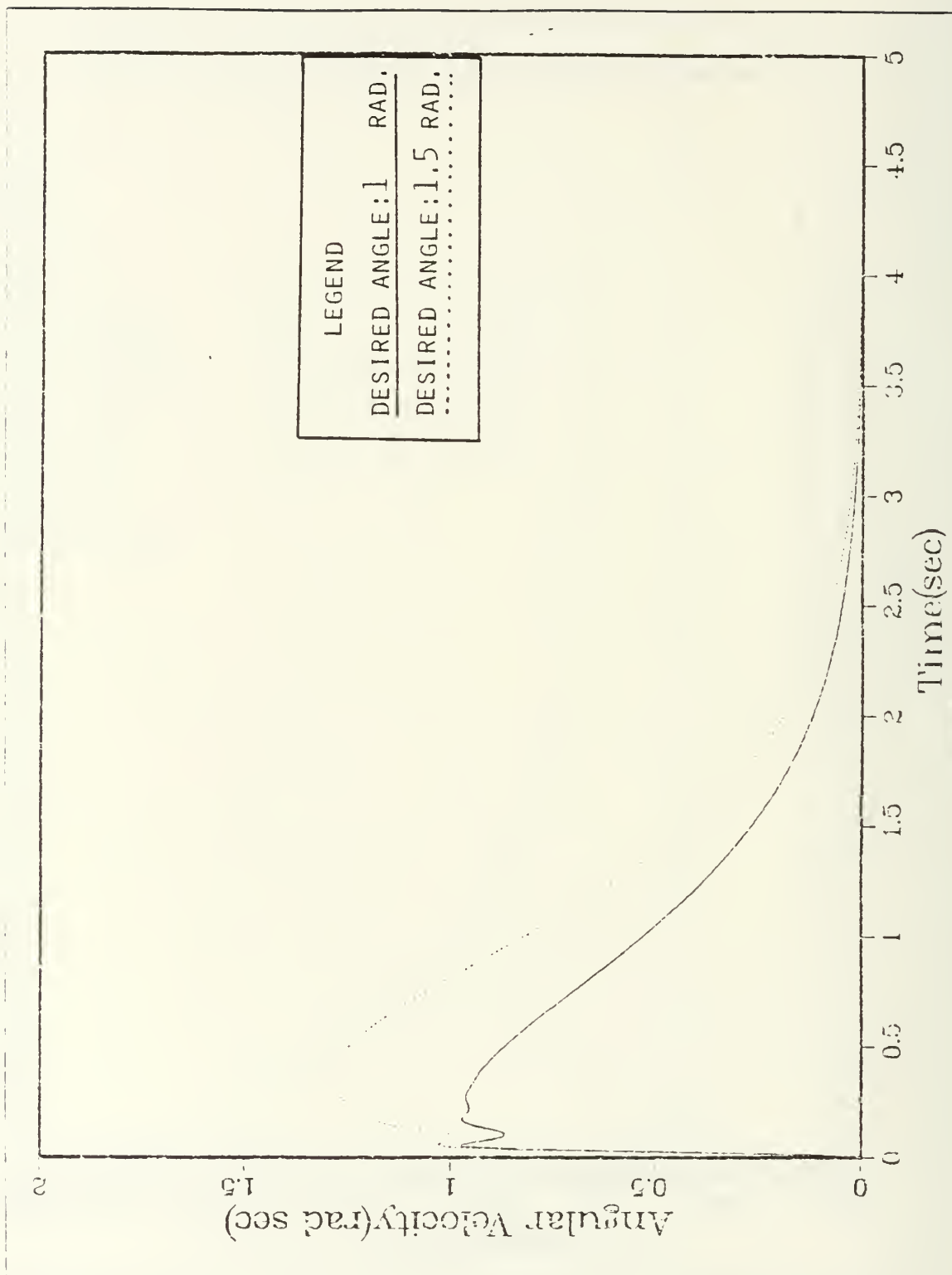
Figure 4.7   Total-Angle Velocities for Various Desired Angles(2.115kg).

42
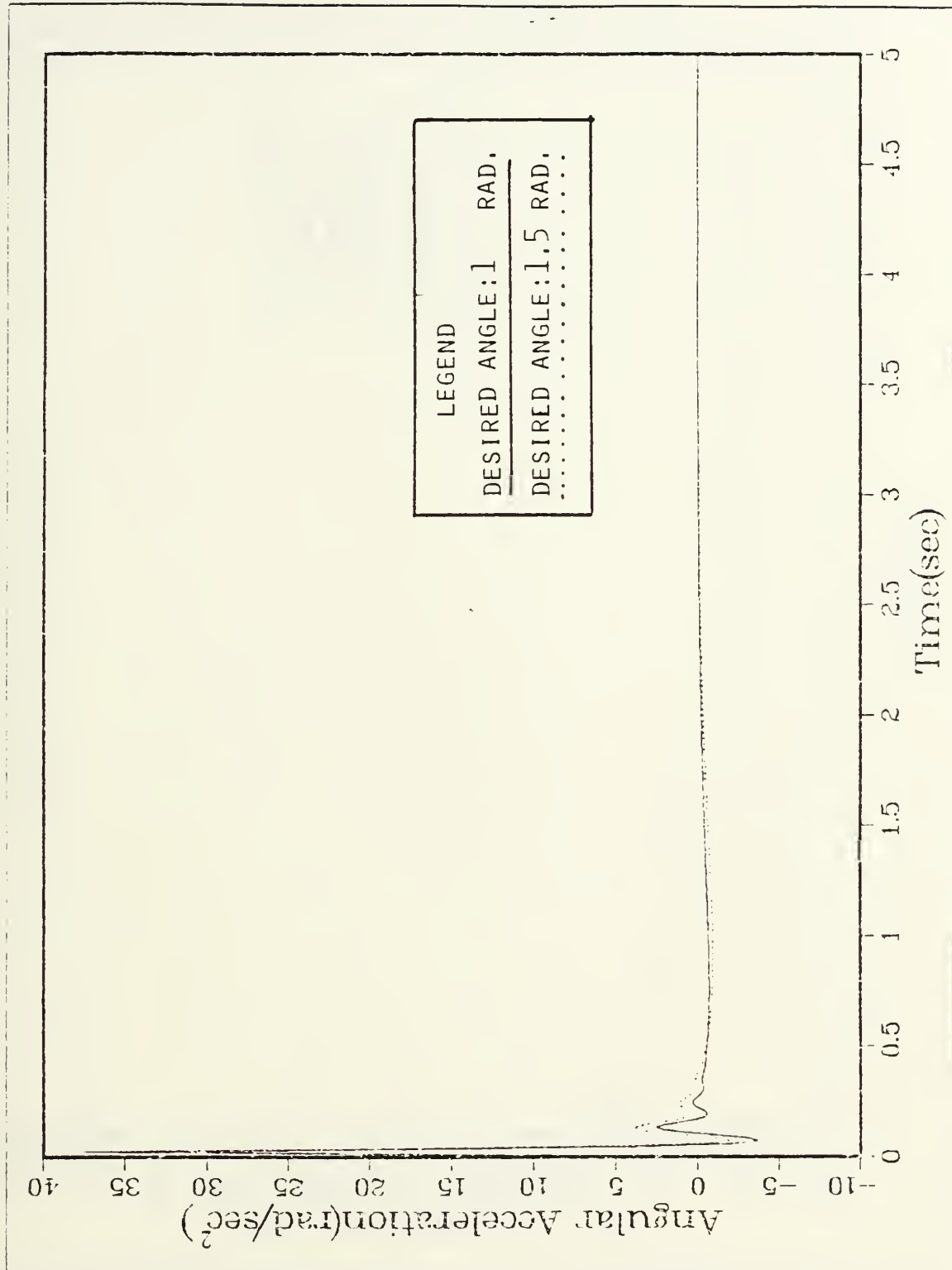
Figure 4.8   Total-Angle Accelerations for Various Desired Angles(2.115kg).
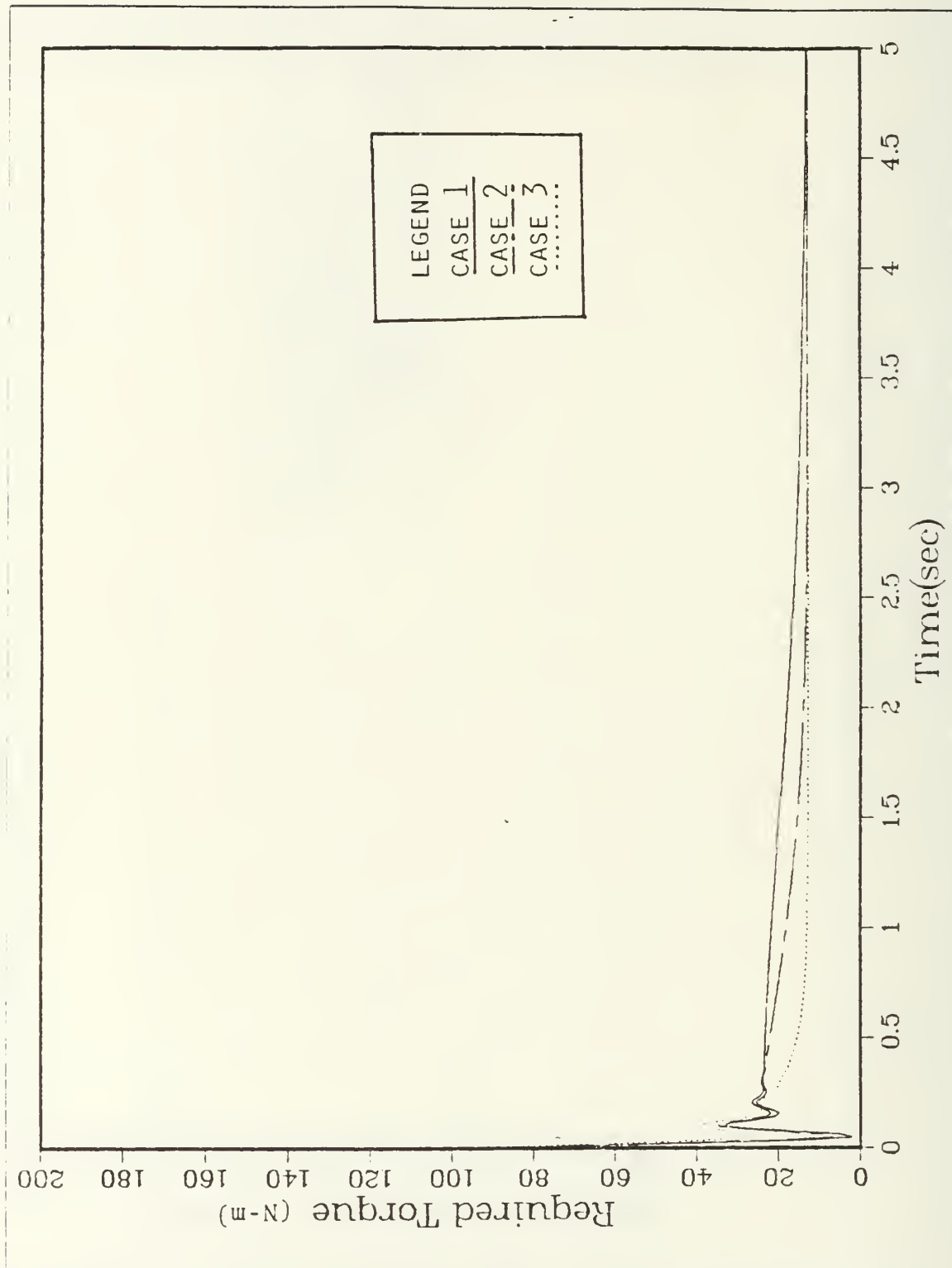
Figure 4.9   Required Torque (0 kg).

44
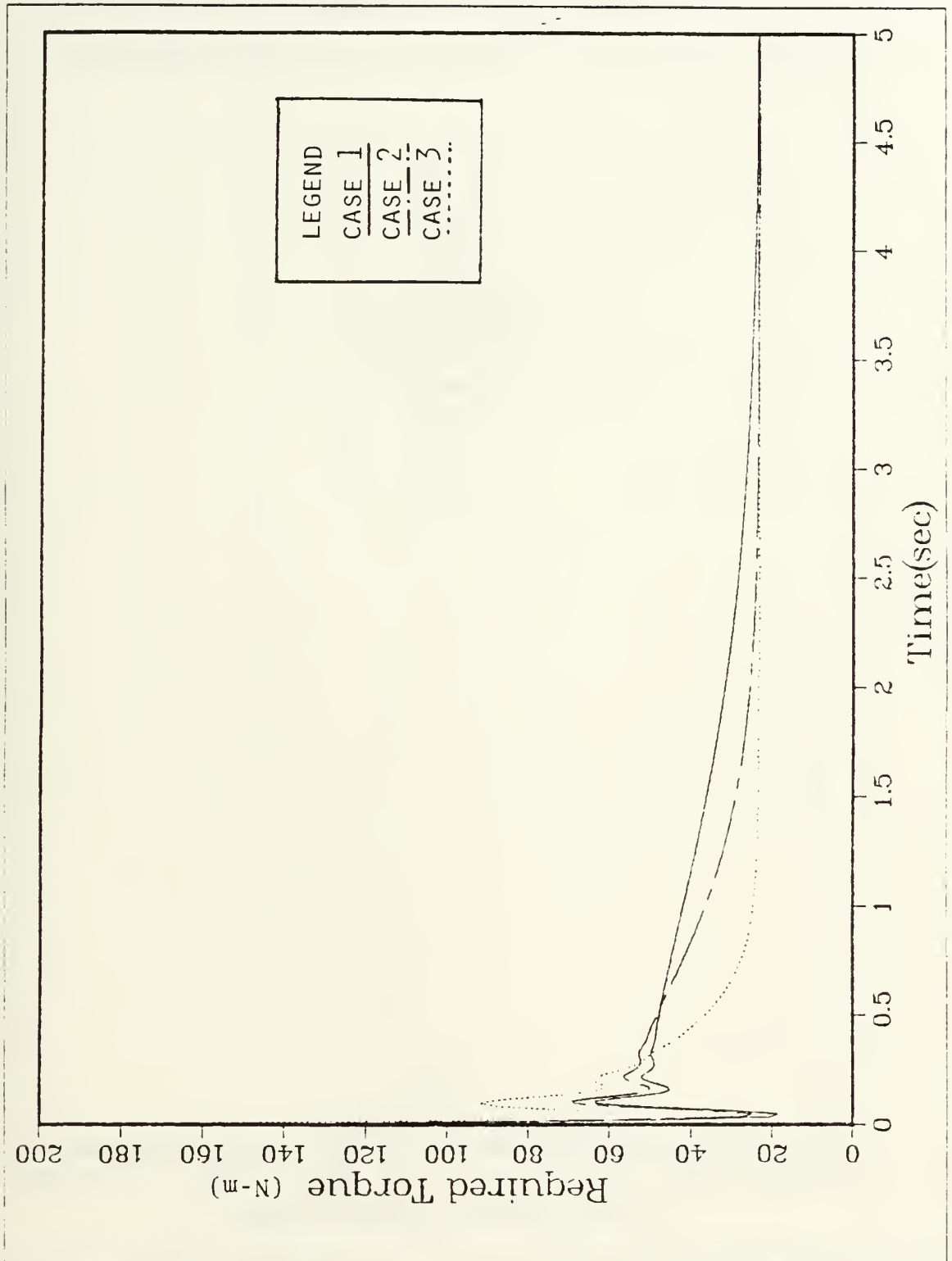
Figure 4.10   Required Torque (2.115 kg).

45
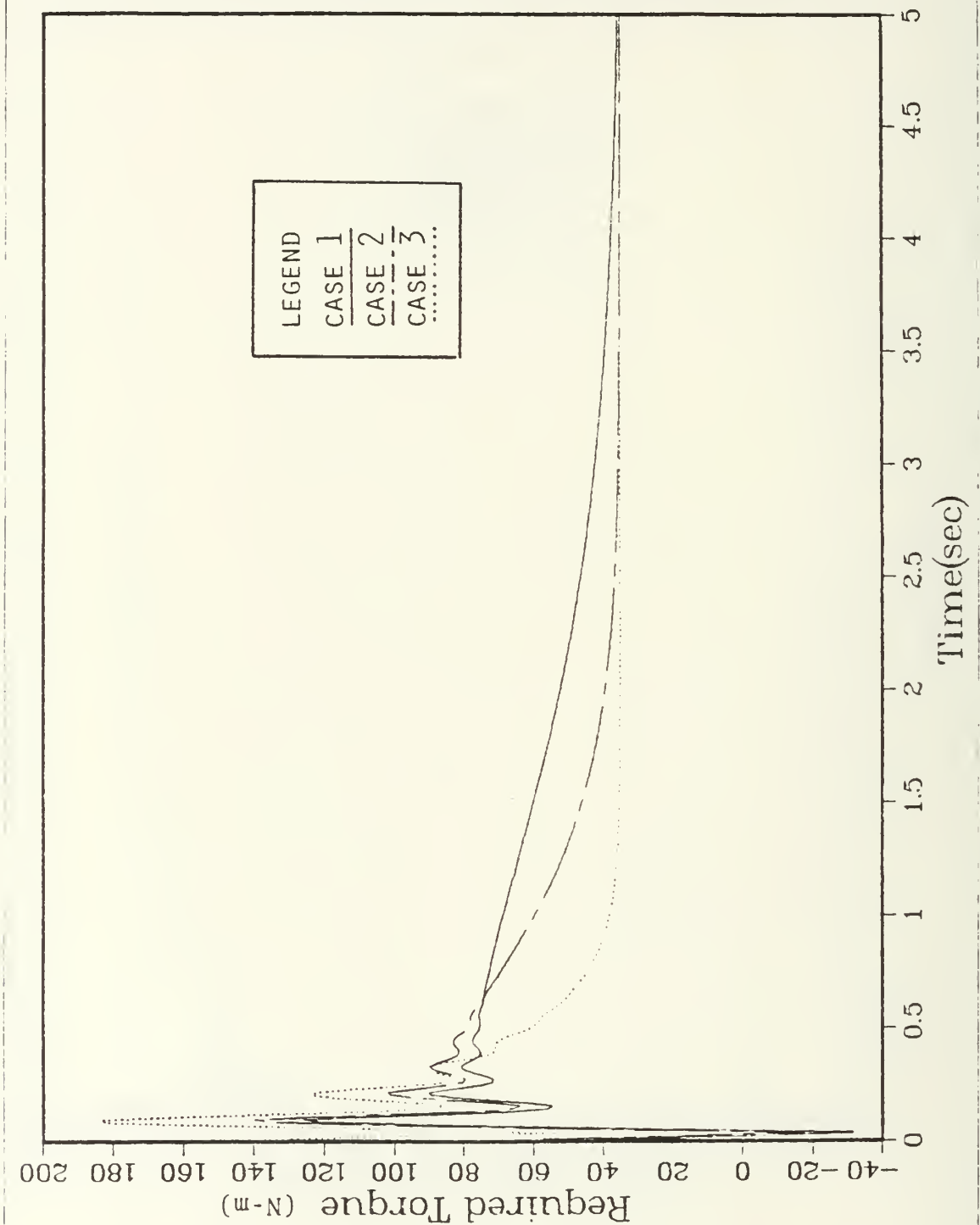
Figure 4.11   Required Torque (4.233 kg).

46
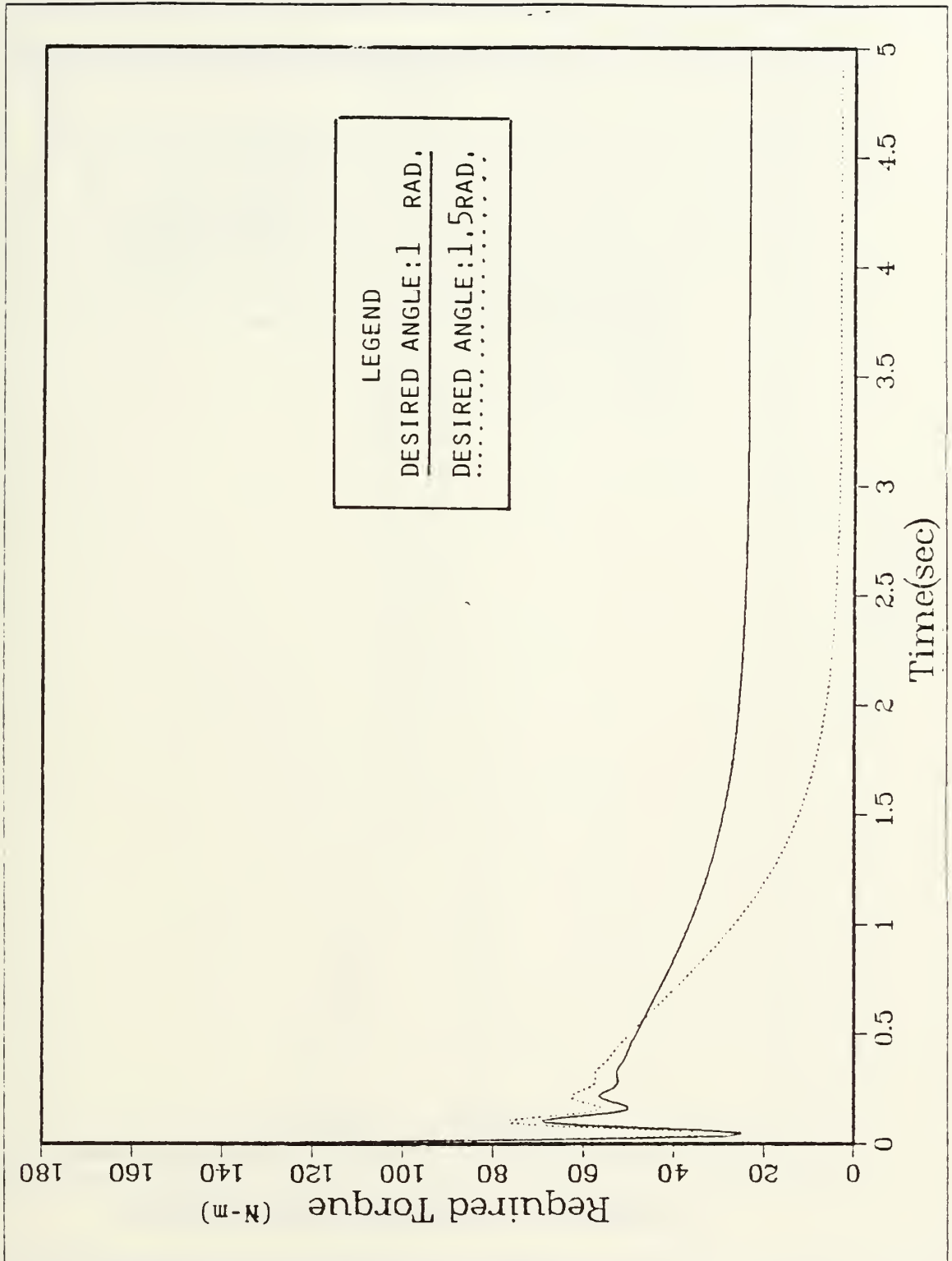
Figure 4.12   Required Torques for Various Desired Angles(2.115kg).

47
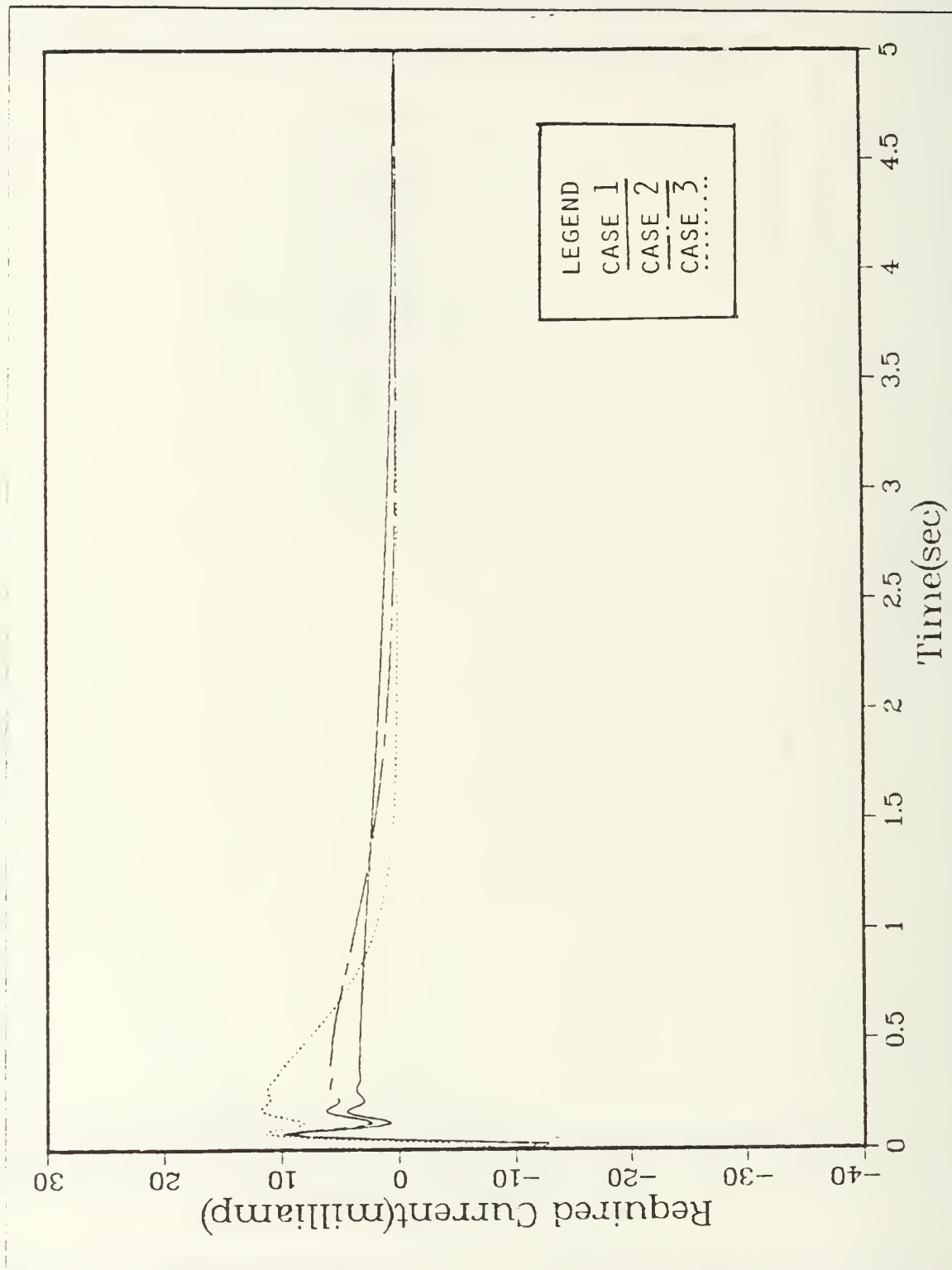
Figure 4.13   Required Current (0 kg).

48
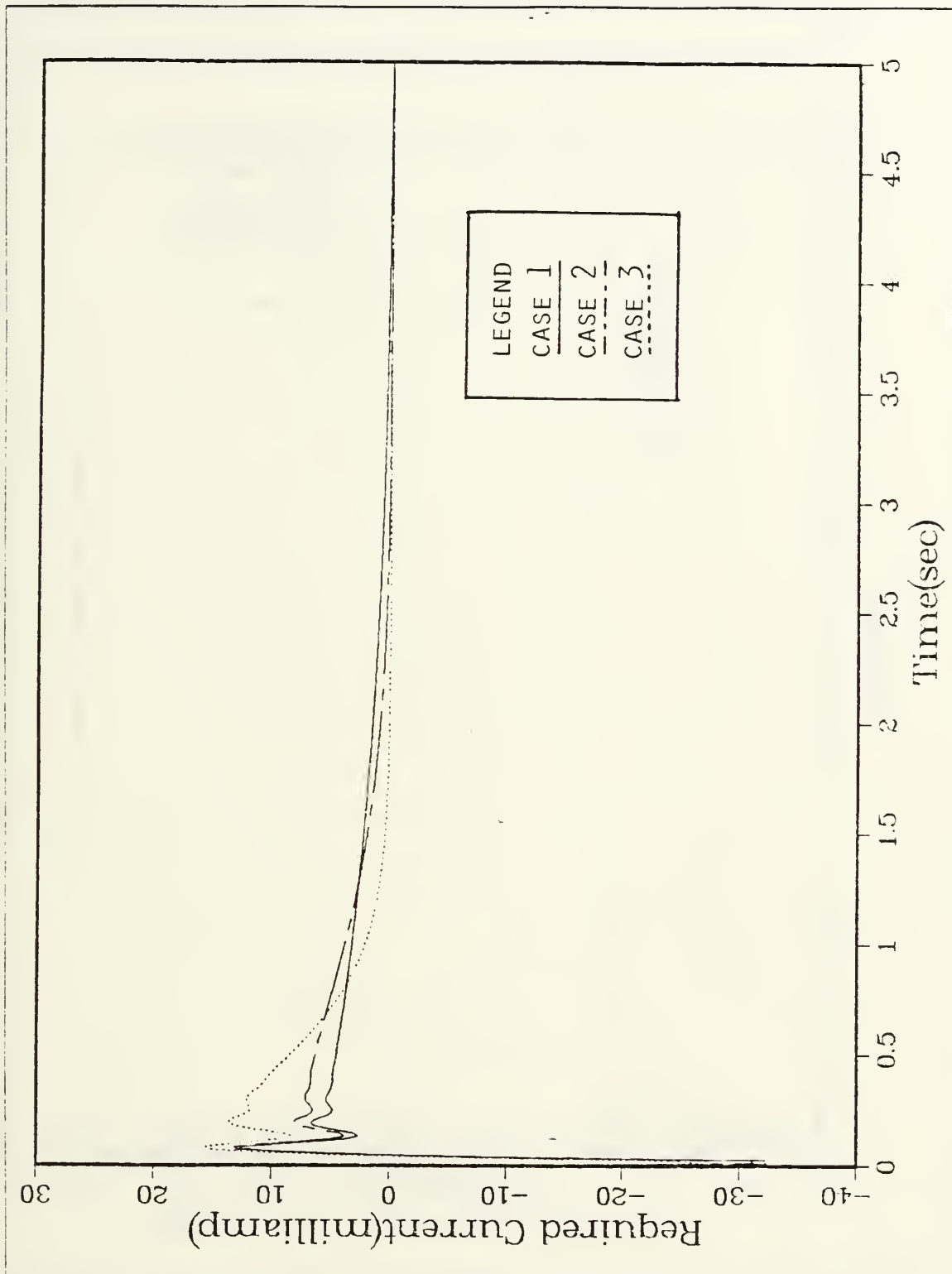
Figure 4.14   Required Current (2.115 kg).

49
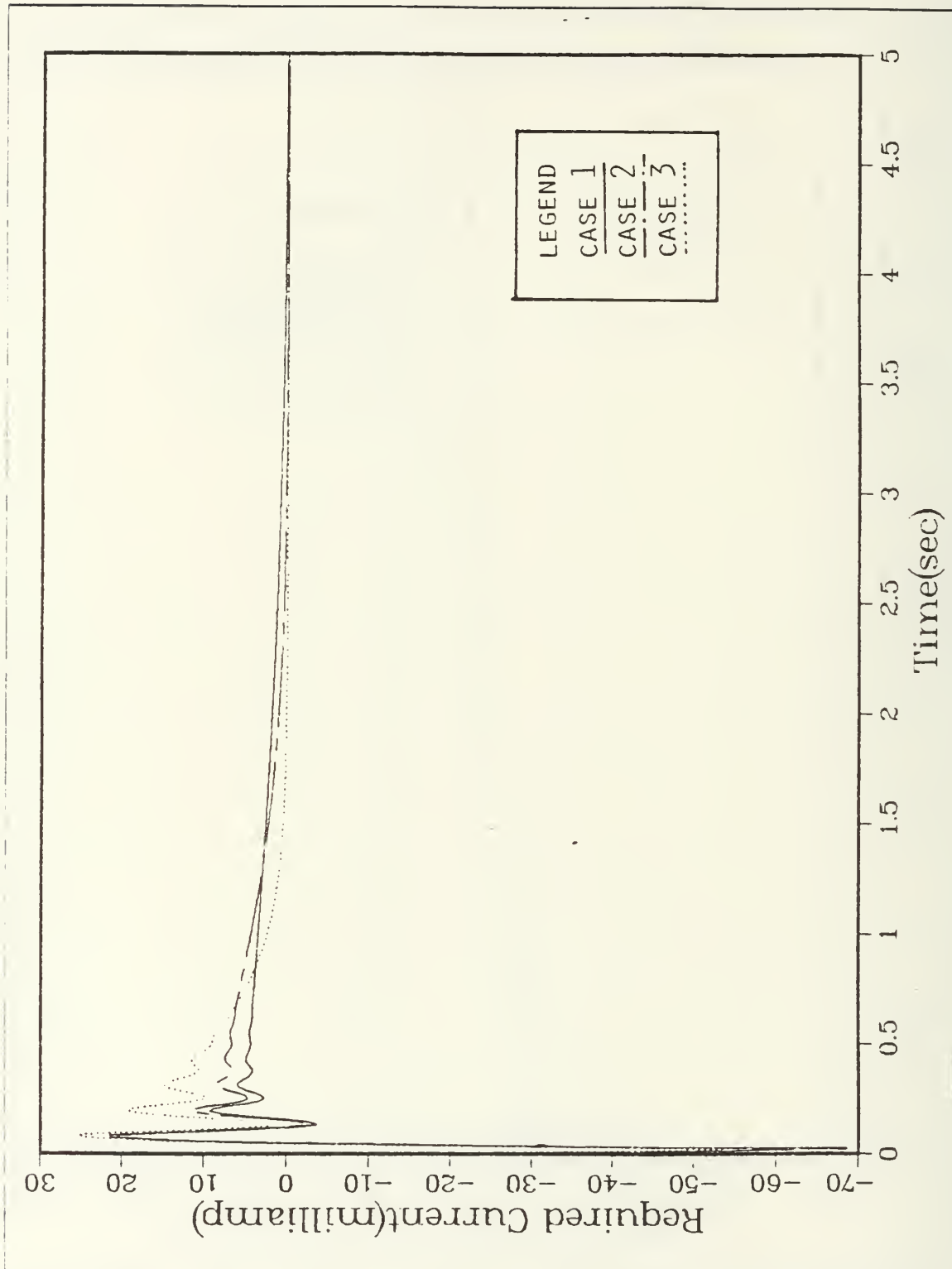
Figure 4.15   Required Current (4.233 kg).

50
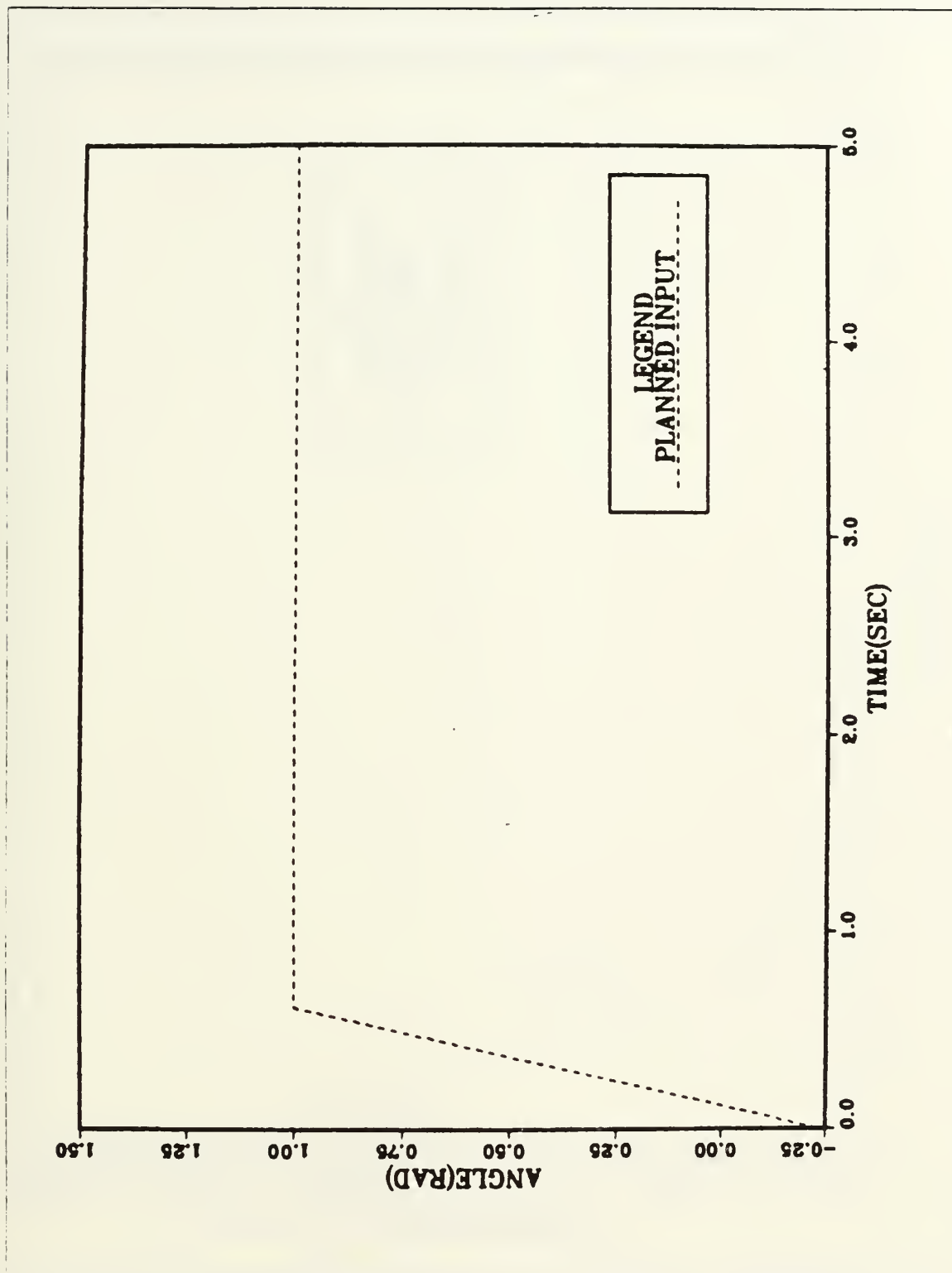
Figure 4.16   A Planned Input of the Desired Angle(4.233 kg).

51
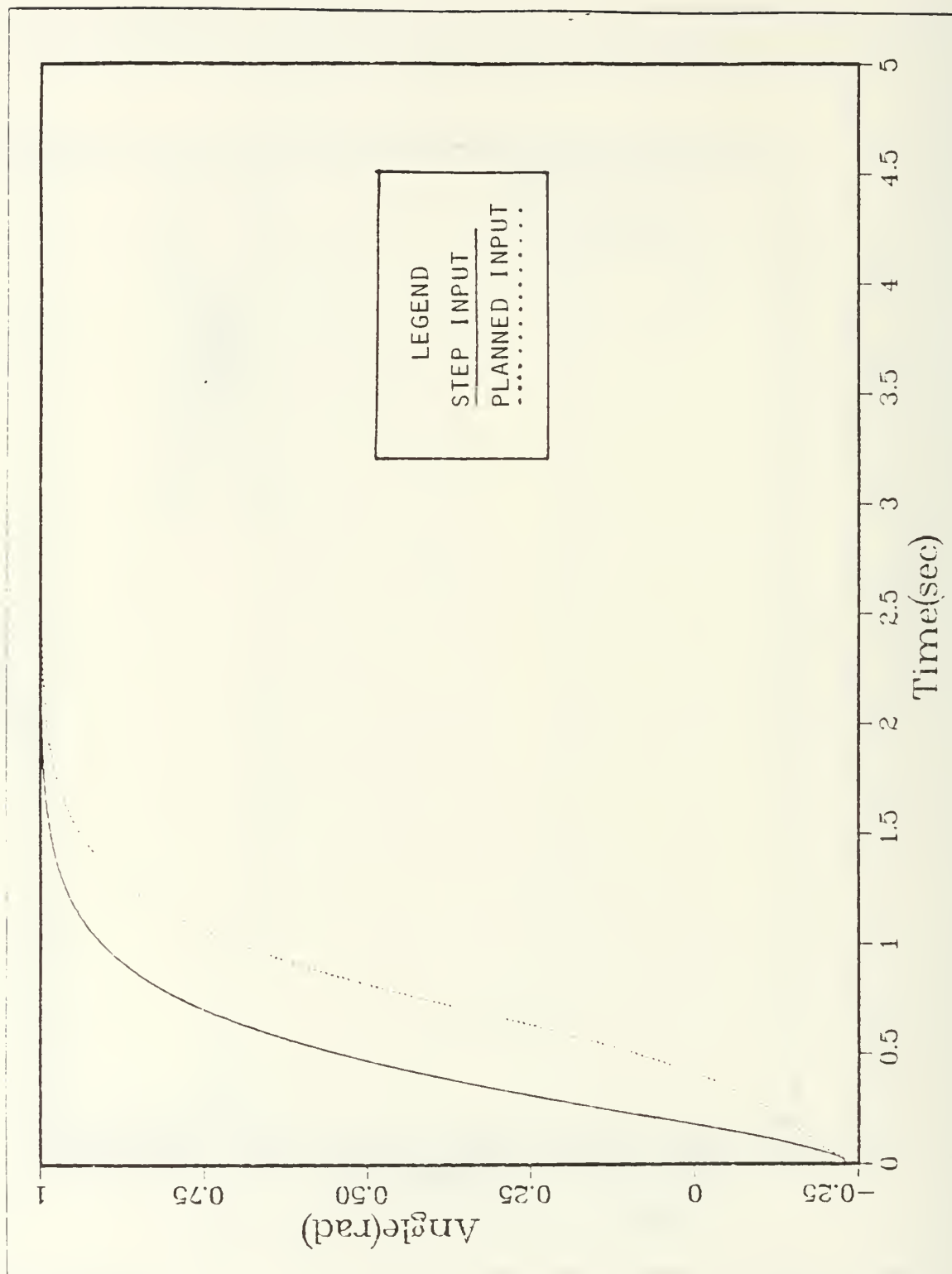
Figure 4.17   Total Angle Responses for Various Input Functions(4.233kg).

52
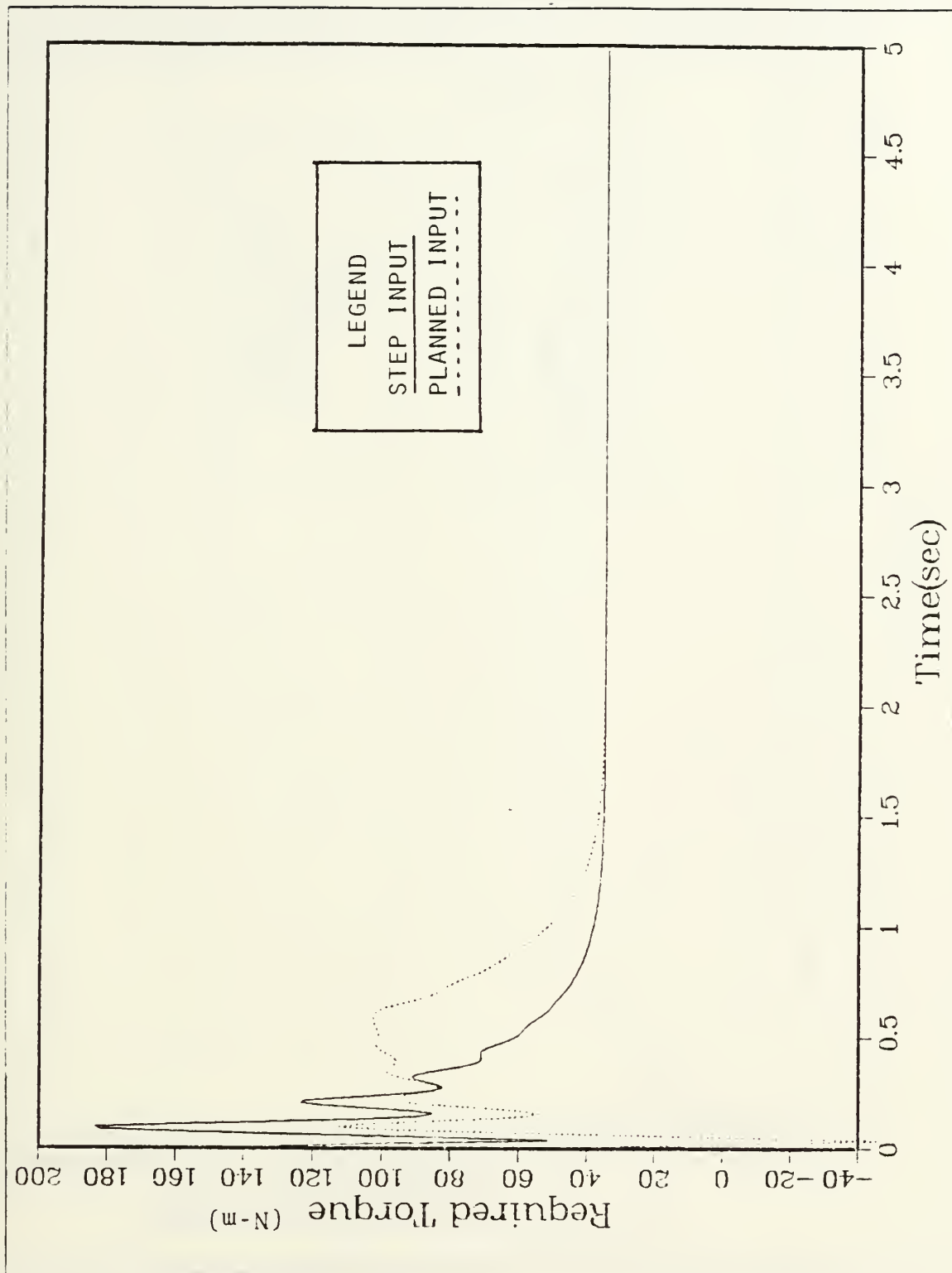
Figure 4.18   Required Torque for Various Input Functions(4.233 kg).

53
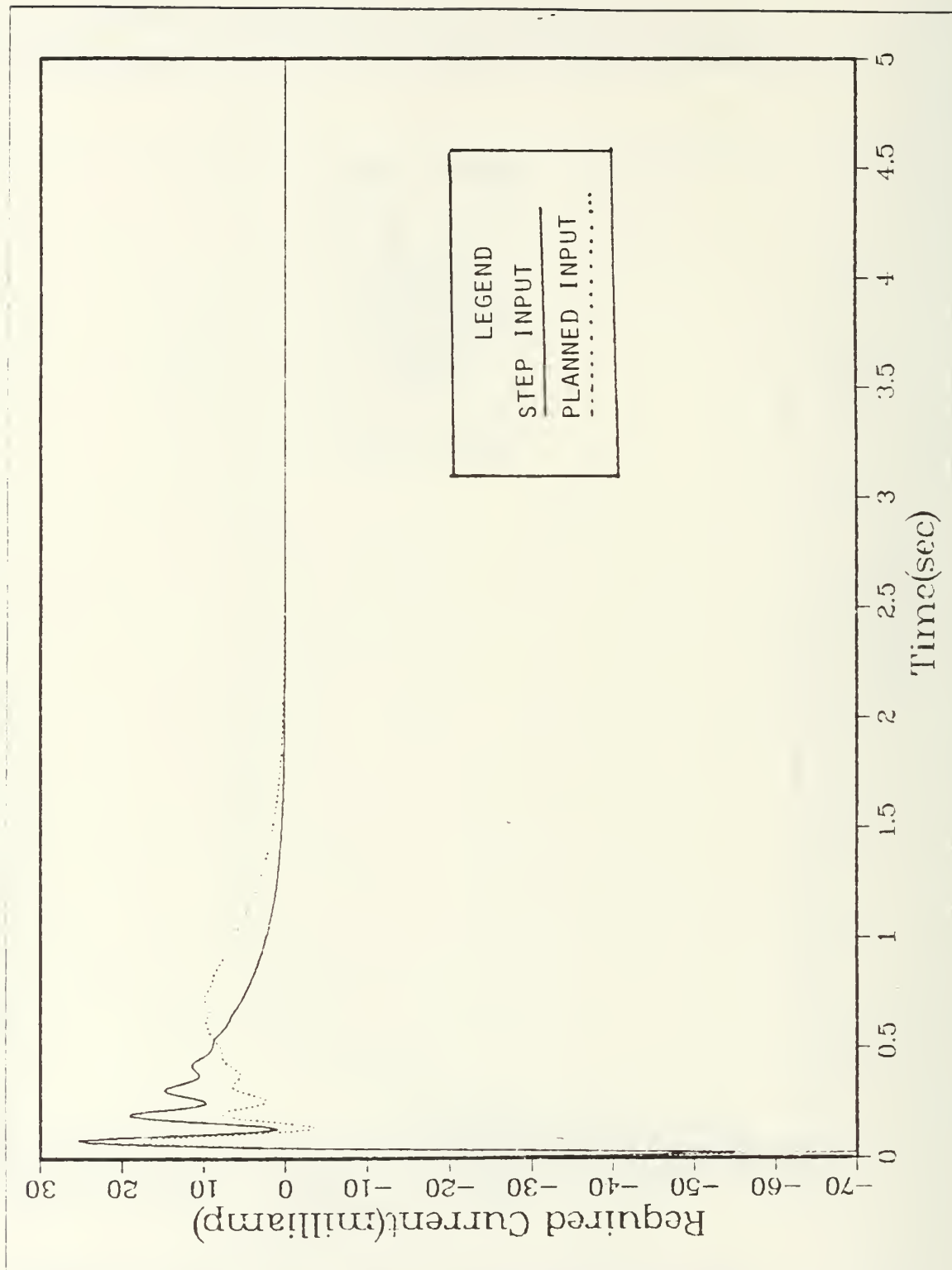
Figure 4.19 Require Current for Various Input Functions(4.233kg).
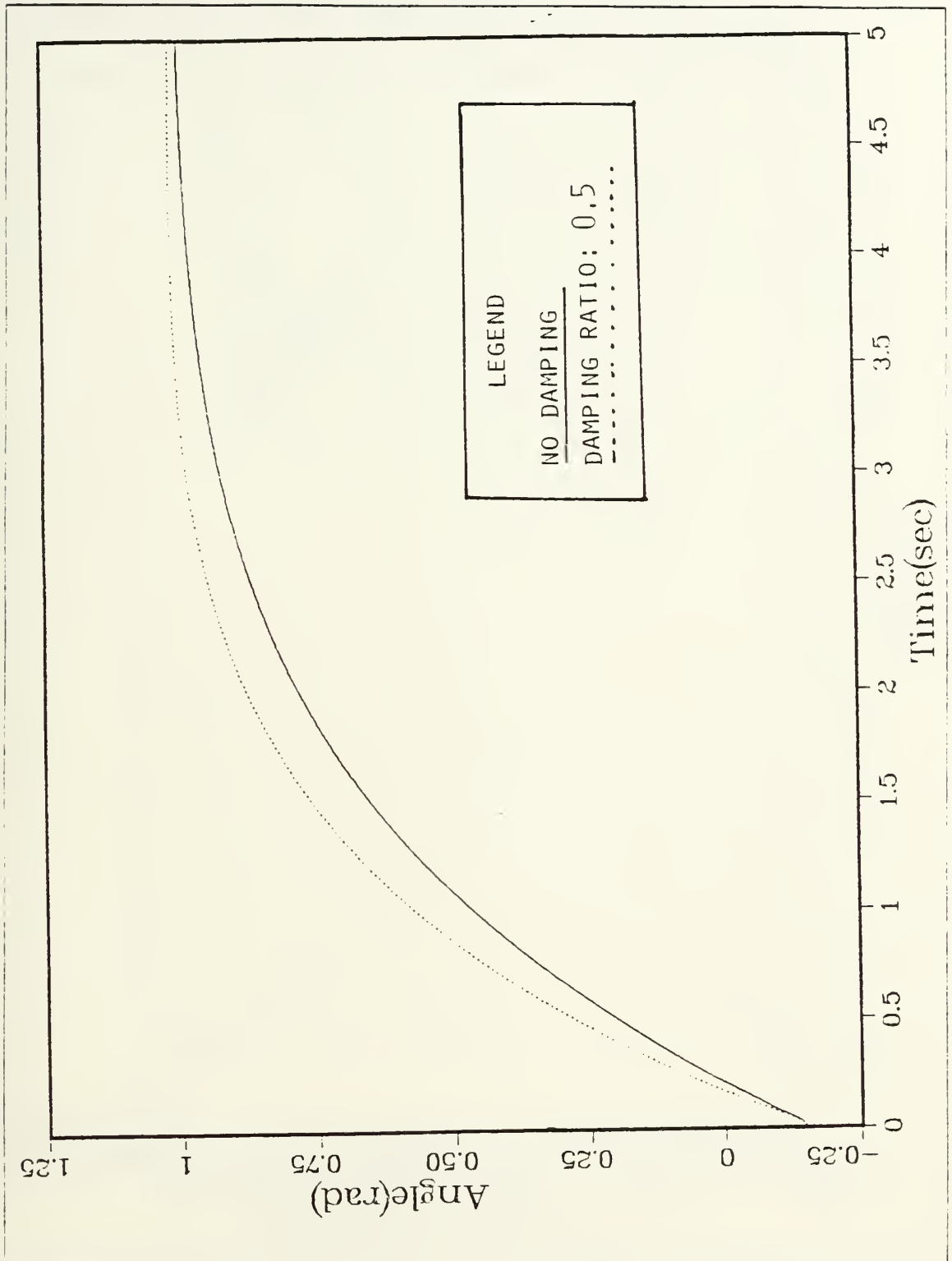
Figure 4.20 Total Angle Response for Various Damping Ratios(no load).
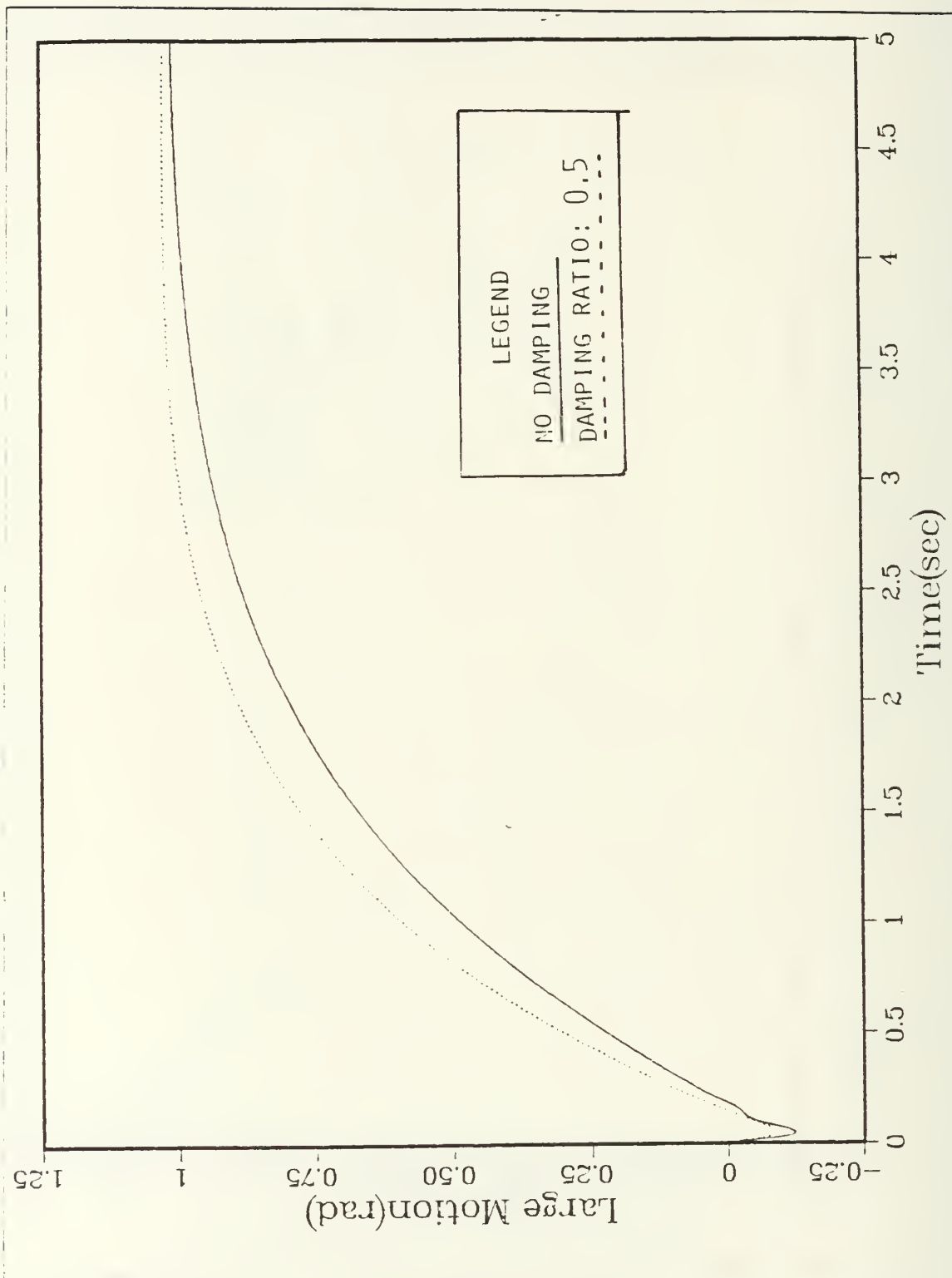
Figure 4.21 Large motion for Various Damping Ratios (no load).
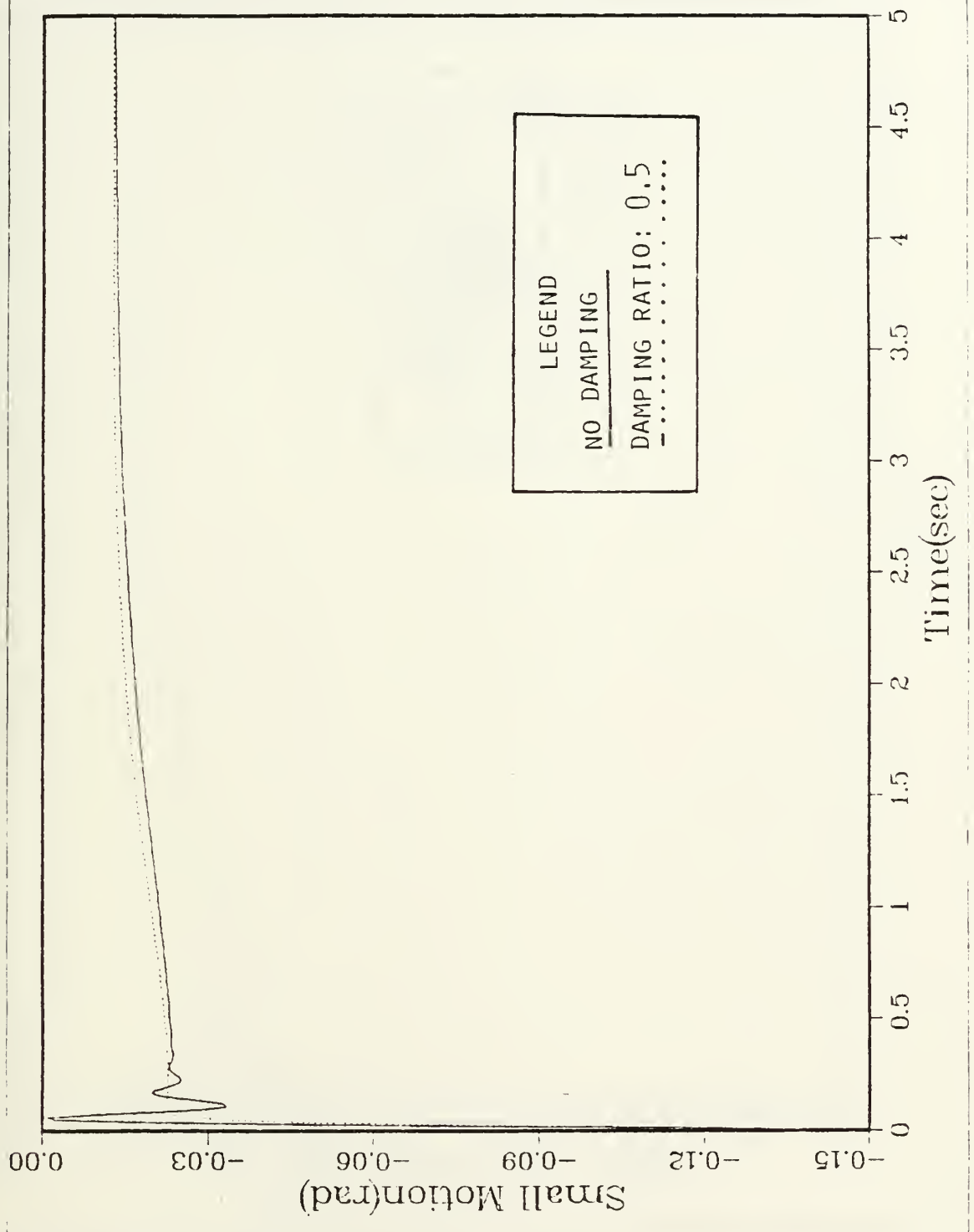
Figure 4.22   Small Motion for Various Damping Ratios (no load).
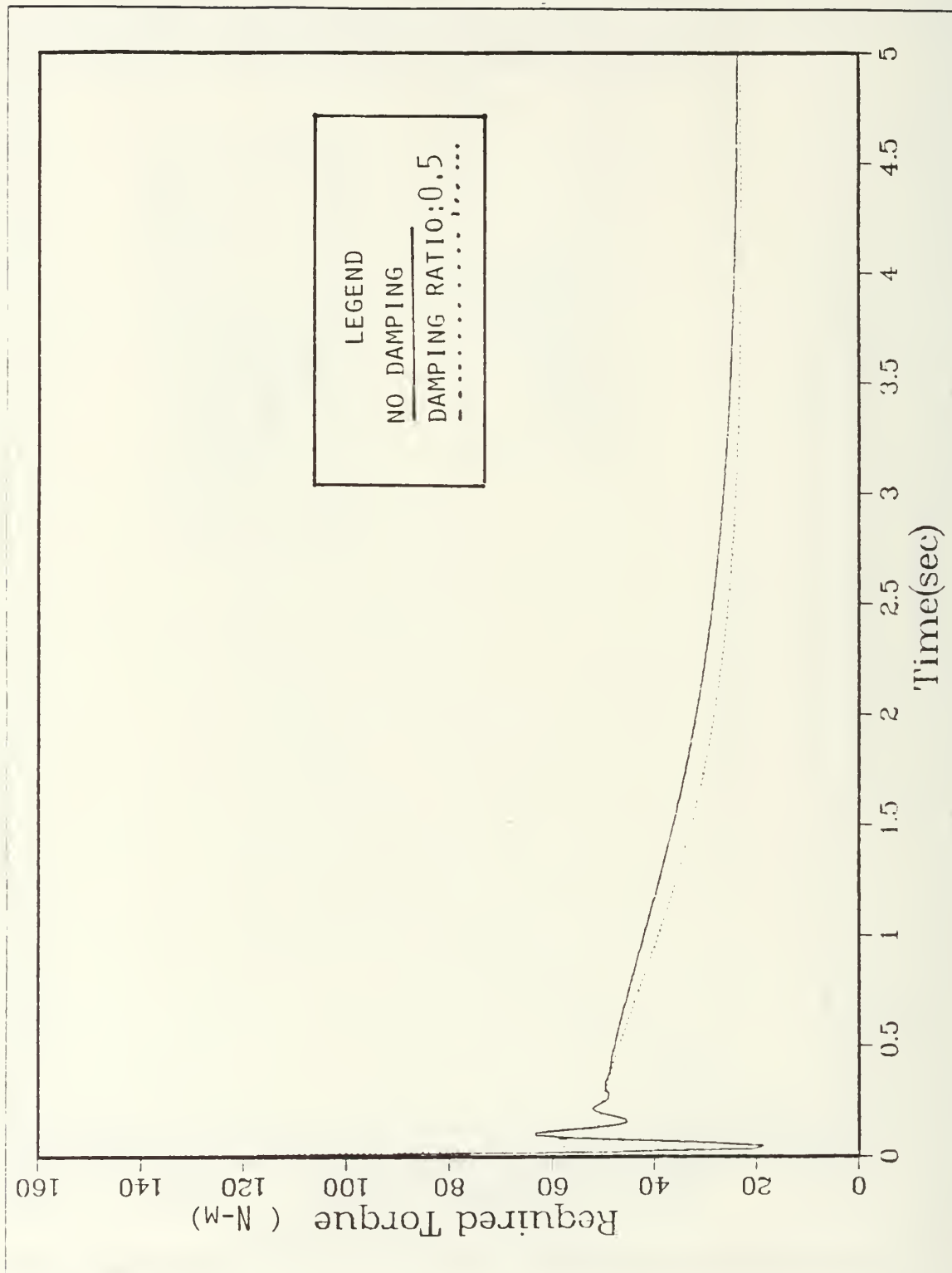
Figure 4.23   Required Torque for Various Damping Ratios (no load).

58
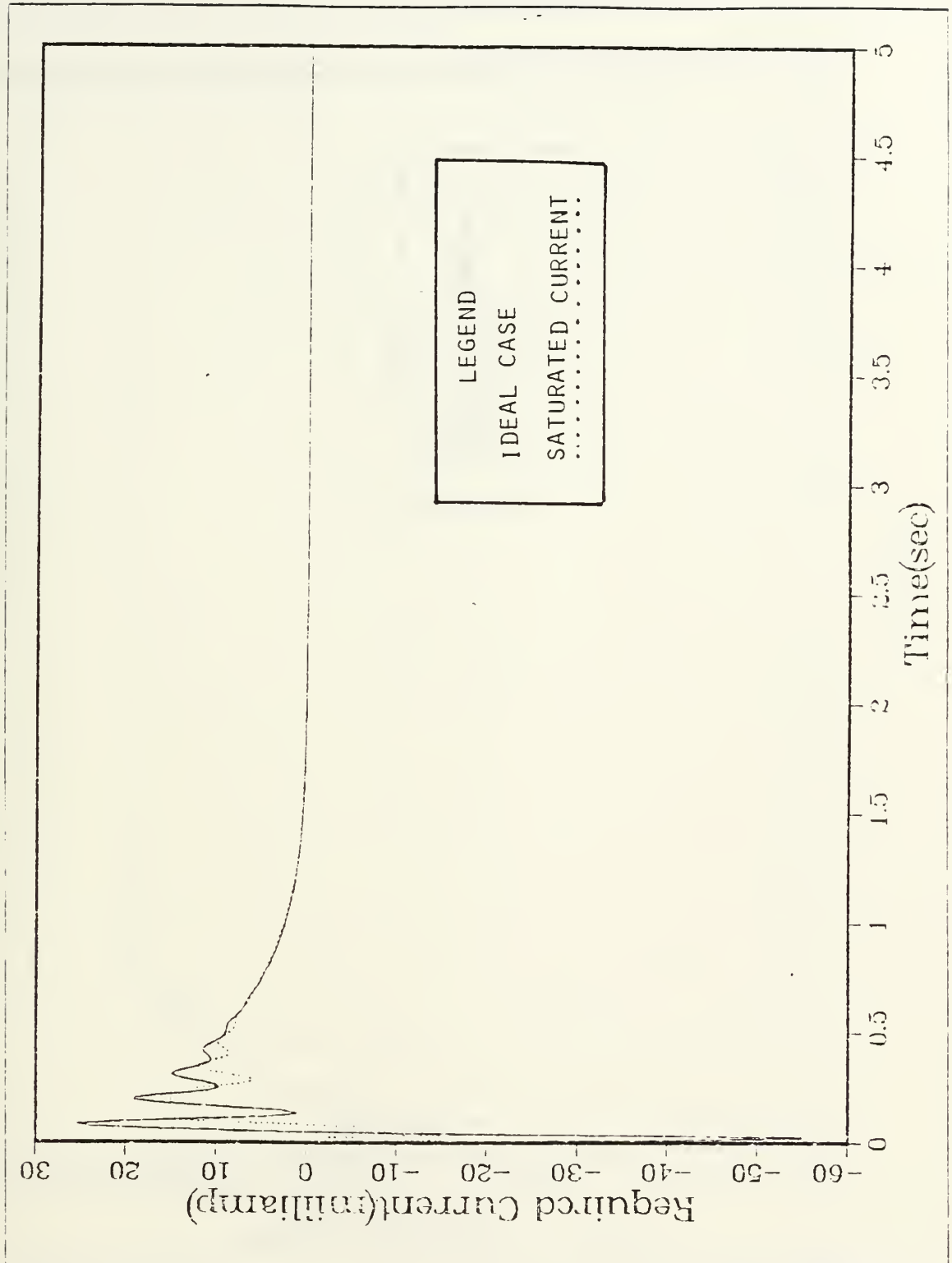
Figure 4.24   Required Currents for the Ideal and Saturated cases(4.233kg).

59
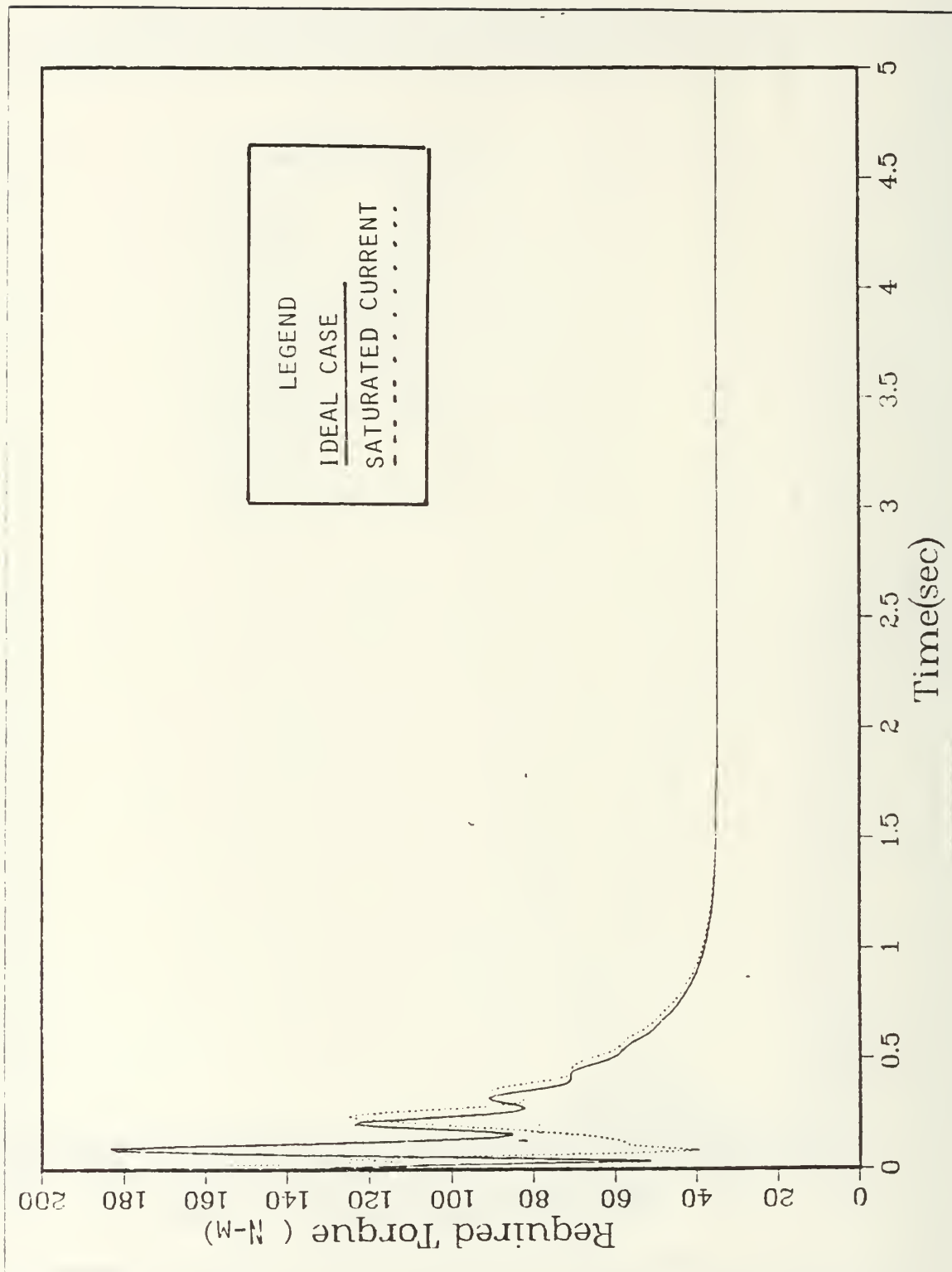
Figure 4.25   Required Torques for the Ideal and Saturated cases(4.233 kg).

60
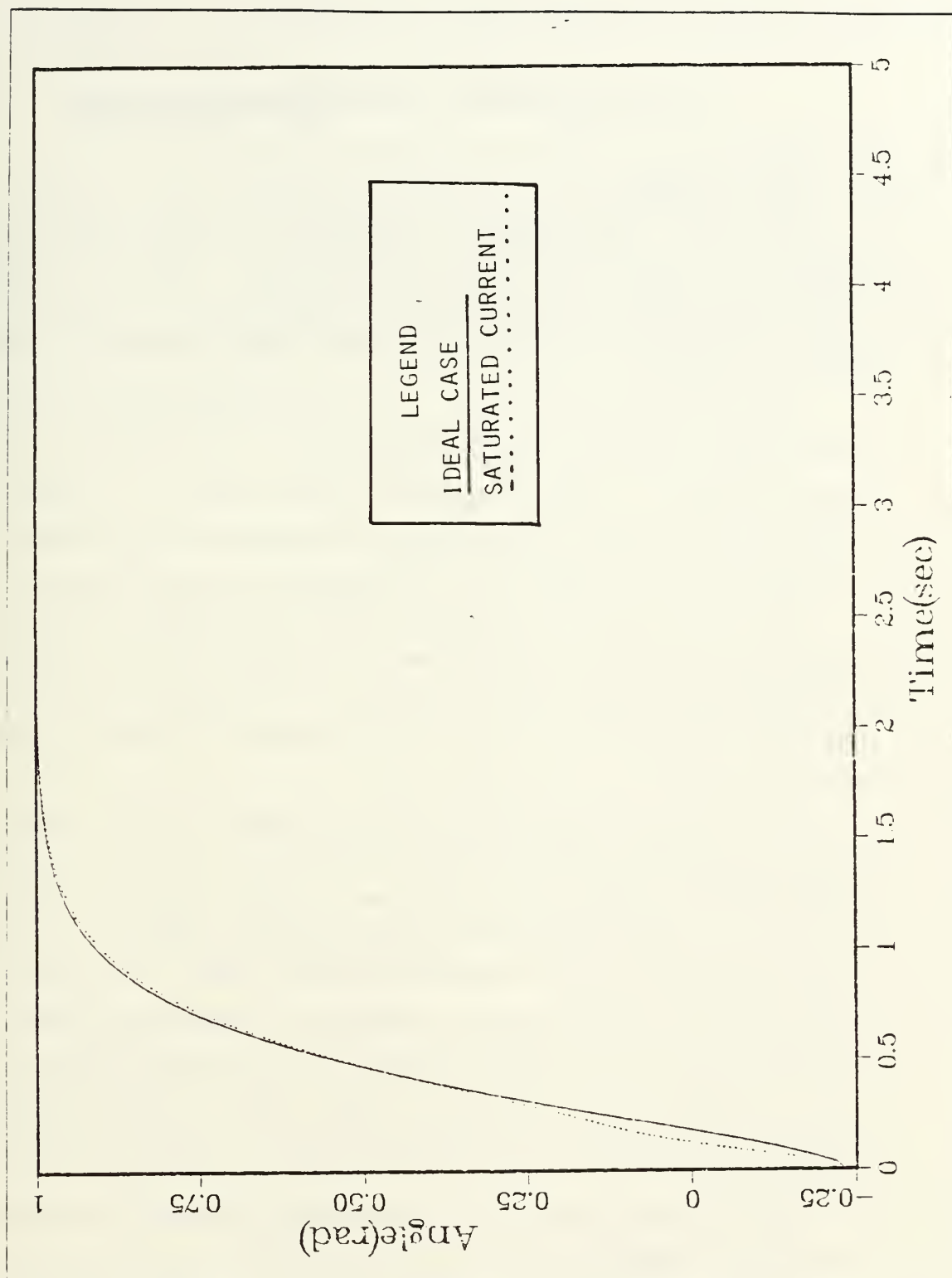
Figure 4.26   Total Angle Responses for the Ideal and Saturated cases(4.233kg).

61

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The purpose of this research is to design a controller for a single-link flexible manipulator using the Equivalent Rigid Link System. The control algorithm is formulated calculating the required torque to control the desired angle of the end-position. Comparisons of the system responses and required torques were made in the different conditions.

The results indicate that system responses and required torques are related to the selected gain values for all loading conditions. The movements of the end-point with higher gain values are faster to achieve the desired position. For different desired positions with the same gain, the end-point movements have the same pattern. These results indicate that non-linear motion is compensated in the control algorithm.

The required torques are increasing to deliver the load and overcome the inertial forces during transient times as gain values are increasing. The system rapidly settles to the steady state. Because of no steady state error, the required torques at the steady state have same values for the same desired positions even though the gains are different.

The maximum of the required torques and required currents are related to the capacity of the electrohydraulic actuator. The selection of size of electrohydraulic actuator affects the weight and cost of the control system. Since more demand is being placed on robot control with higher speeds and smaller actuators, it is important to reduce the maximum required torques and required currents. The application of the planned input of the desired angle as a combination of ramp and step functions lead a noticeable reduction of the maximum required torques and required currents. However, slower movements of the end-point are yielded during the transient period. The maximum required torques and required currents can be adjusted by the slope of the ramp function.

The results also indicate that the system damping provides a noticeable reduction of the structural vibration.

In addition, when the electrohydraulic actuator is saturated, the movement of the end-point is slower than the ideal case without saturation. The simulation serves an

useful method to run the physical plant within the operating limit of the selected electrohydraulic actuator.

B.   RECOMMENDATIONS

The long-term goal of this research is to design a controller being capable of achieving end effector positioning accuracy and stable movements. Simulation studies need to be continued for flexible multi- link system design using the ERLS model. In addition, the effects of the robustness and spillover problems need to be studied. Extention of the controller design and implementation to the multi-link of the ERLS model is eventually needed if the advantages of flexible manipulator is to be realized in practical industrial application.

APPENDIX A
DERIVATION OF THE EQUATIONS OF MOTION FOR THE SINGLE - LINK FLEXIBLE MANIPULATOR

The motion of the flexible link manipulator has been separated into large motions and small motions by the ERLS. The generalized coordinates are chosen to describe the large motions by the joint variable of ERLS and to describe the small motions by nodal displacements of link. The two generalized coordinates chosen are the rigid body rotation, θ, and the nodal displacement, $U(0)$ which can be measured at the end of the link. The following are the two sets Lagrange equations used to develop the equations of motion;

$$d/dt\{ \partial KE/\partial \dot{\theta}\} - \partial KE/\partial \theta + \partial PE/\partial \theta = F$$

$$d/dt\{ \partial KE/\partial \dot{U} \} - \partial KE/\partial U + \partial PE/\partial U = 0$$

where

    KE - kinetic energy

    PE - potential energy

    θ -   large motion joint variable measured between the ERLS link and
                  the absolute x axis.

    U   - small motion displacement and slope

    F   - generalized force for large motion

The total kinetic energy is due to kinetic energy of the link motion $(KE)_L$ and that of the actuators $(KE)_a$;

$$KE = (KE)_L + (KE)_a$$

The actuator are much more rigid than the link, the actuator is treated as a rigid body and all kinetic energy terms are calculated separatly. Generalized forces include any applied force and damping forces. The expressions utilized for the determination of the kinetic energy of the system are as follows;

$$KE(arm) = 1/2 \int_{\substack{arm \\ volume}} \mu \, \dot{R}^t \{\dot{R}\} \, dv$$

$$KE(\text{load}) = 1/2 \text{ Trace} \int_{\substack{\text{load} \\ \text{volume}}} \mu_L \, \dot{R}_L^{\ t} \{\dot{R}_L\} \, dv$$

$$KE(\text{rotor}) = 1/2 \text{ Trace} \int_{\substack{\text{rotor} \\ \text{volume}}} \mu_r \, \dot{R}_r^{\ t} \{\dot{R}_r\} \, dv$$

The absolute position vector of the arm is determined from the following transformation;

$R = W(r + D)$

W = the 3×3 transformation matrix of function of theta, θ, only.

r = the 3×1 local position vector of the arm measured from the coordinate system whose origin is the end of the ERLS link.

d = the 3×1 deformation vector that only includes the transverse displacement, derivation of this vector is discussed in chapter2.

μ = the mass density of the steel arm.

The absolute position vector of load is determined from the following transformation;

$R_L = W \, D_L \, r_L$

D_L = the 3×3 transformation matrix due to the local deformation of the arm tip.

r_L = the 3×1 local position vector for the load

μ_L = mass density of the steel load

The absolute position vector of the hydraulic actuator is given by the following transformation;

$R_R = A_R \, r_R$

A_R = the 3×3 transformation matrix due to the large motion rotation of the rotor

r_R = the 3×1 local position vector for the rotor

μ_R = the mass density of the actuator rotor, aluminum

Potential energy of the flexible arm comes from strain energy $(PE)_g$ of the structure due to deformation and the gravitational energy $(PE)_g$ of the system. The strain energy is produced by bending, tortion and extention. Along modeling, the lateral shear deflections are neglected so that the system energy of bending is due to flexure only. The expressions used for the determination of potential energy are as followes;

$$PE_d = 1/2 \int_{\substack{link \\ length}} E \, I_{zz} \, (v'')^2 \, dx$$

$$PE_g = - \int_{\substack{link \\ volume}} \mu \, r^t \, g \, dv$$

$$PE_{gL} = - \int_{\substack{load \\ volume}} \mu_L \, r^t \, g \, dv$$

where

$E \, I_{zz}$ = the flexural rigidity in the z direction, perpendicular to
    the plane of motion

v'' = second derivative of the transverse displacement V with respect
    to the local x coordinate direction expressed interms of $V(0)$

g = gravitational acceleration vector

The following definitions for the inertia terms are utilized to simplify the computations and the resultant expressions in the equations motion;

I_L = the 3 × 3 inertia matrix of the load

   = $\int_{\substack{load \\ volume}} \mu_L \, r_L \, r_L{}^t \, dv$

I_R = the 3 × 3 inertia matrix of the rotor

   = $\int_{\substack{rotor \\ volume}} \mu_R \, r_R \, r_R{}^t \, dv$

$$I111(W_\theta{}^t, W_\theta) = \int_{\substack{link \\ volume}} \mu \, r^t \, W_\theta{}^t \, W_\theta \, r \, dv$$

$$I112(W_\theta{}^t, W) = \int_{\substack{link \\ volume}} \mu \, r^t \, W_\theta{}^t \, W \, \Psi \, r \, dv$$

$$I121(W^t, W_\theta) = \int_{\substack{link \\ volume}} \mu \, \Psi^t \, W_\theta{}^t \, W \, dv$$

$$I122(W^t, W) = \int_{\substack{link \\ volume}} \mu \, \Psi^t \, W^t \, W \, \Psi \, dv$$

$$I122(W^t, \ddot{W}_R) = \int_{\substack{link \\ volume}} \mu \, \Psi^t \, \ddot{W}^t \, W_R \, \Psi \, dv$$

$$I122(W_\theta{}^t, \dot{W}) = \int_{\substack{link \\ volume}} \mu \, \Psi^t \, W_\theta{}^t \, \dot{W} \, \Psi \, dv$$

$$I122(W_\theta{}^t, W_\theta) = \int_{\substack{link \\ volume}} \mu \, \Psi^t \, W_\theta{}^t \, W_\theta \, \Psi \, dv$$

$$I_{yy} = \int \mu_L \, y^2 \, dv$$
$$I_{xx} = \int \mu_L \, x^2 \, dv$$

where

Ψ = the 3×2 link shape matrix

W_θ = derivative of the 3×3 matrix W with respect to the joint variable θ

$\ddot{W}_R$ = result of the simplification of the second time derivative of
the transformation matrix W and is termed the residual acceleration.

The following definitions are utilized to simplify the computations and the expressions used for computer coding of the equations of motion;

67

$$K_{11} = \int_{\substack{\text{link}\\\text{length}}} \Gamma^t \, C \, \Gamma \, dx$$

$$H_{11} = \int_{\substack{\text{link}\\\text{volume}}} \mu \, r^t \, dv$$

$$H_{21} = \int_{\substack{\text{link}\\\text{volume}}} \mu_l \, \Psi^t \, dv$$

$$H_{41} = \int_{\substack{\text{link}\\\text{volume}}} \mu_L \, r_L{}^t \, dv$$

Γ = the second derivative of the shape function matrix

C = the 3 × 3 flexural rigidity matrix including only $E\,I^{zz}$

The actual derivation of the equations of motion requires detailed, tedious substitution of the expressions for the kinetic energy and potential energy into the Lagrange equations, and the Lagrangian formulation yields two sets of equations, One set describes the large motions and the other set describes the small motions. These two sets of equations are non- linear, coupled, second - order, ordinary differential equations represented as follows;

$$M_{qq} \, \ddot{\theta} + M_{qn} \, \ddot{U} = F_q + \text{TORQUE}$$

$$M_{nq} \, \ddot{\theta} + M_{nn} \, \ddot{U} + K_n \, U = F_n$$

where

M_{qq} = 1 × 1 effective inertia matrix for large motions

M_{qn} = 1 × 2 coupled inertia matrix of the small motion effect on large motions

M_{nq} = 2 × 1 coupled inertia matrix of the large motion effect on small motions

M_{nn} = 2 × 2 effective inertia matrix for small motions

K_n = 2 × 2 stiffness matrix for small motions

F_q = 1 × 1 load vector for large motions

F_n = 2 × 1 load vector for small motions

θ = generalized coordinate of the large motions

U = 2 × 1 generalized coordinate of the deformations representing the small motions-the displacement and slope.

The coefficients for the terms are defined as follows;

$$M_{qq} = I111(W_\theta{}^t, W_\theta) + U^t I122(W_\theta{}^t, W_\theta) U$$
$$+ \text{Trace} (W_\theta D_L I_L D_L{}^t W_\theta{}^t + A_R I_R A_R{}^t)$$

$$M_{qn} = I112(W_\theta{}^t, W_\theta) + ((M_L L + M_x), (M_x L + I_{xx} + I_{yy})$$

$$F_q = 2 U^t I122(W_\theta{}^t, W) U + H_{11} W_\theta{}^t +$$
$$U^t H_{21} W_\theta{}^t g - \text{Trace} (W_\theta D_L I_L D_L{}^t W_r$$
$$+ 2 W_\theta D_L I_L D_L{}^t W$$
$$+ A_R\theta I_R \ddot{A}_{RR}{}^t) + H_{11} D_L{}^t W_\theta g + T$$

$$M_{nq} = (\text{Trace} (W_\theta D_L I_L D_{L11}{}^t W^t),$$
$$\text{Trace}(W_\theta D_L I_L D_{L12}{}^t W^t + I121(W^t, + I121(W^t, W_\theta)$$

$$M_{nn} = I122 (W^t, W) + M_L (\ddot{v}(0)) + (I_{xx} + I_{yy}) \times (\ddot{\varphi}(0))$$

$$K_n = K^{11} + I122 (W^t, W_R)$$

$$F_n = H_{21} W^t g - (\text{Trace} (W_R D_L I_L D_{L11} W^t), + 2 W D_L I_L D_{L11}{}^t W^t),$$
$$\text{Trace}(W_R D_L I_L D_{L12}{}^t W^t + 2 W D_L I_L D_{L12}{}^t W^t))$$
$$+ H_{41} D_{L11}{}^t W^t g , H_{41} D_{L12}{}^t W^t g$$

where

L = the arm length

M_L = the mass of the load

M_x = the first moment of the load with respect to the local y axis.

T = the externally applied torque

A_{RR} = a result of simplification of the second time derivative of the transformation matrix A_R and is termed D_{12} residual acceleration.

$A_R\theta$ = a derivative of A_R with respect to the joint variable theta.

D_{L11}, D_{L12} = the derivatives of the arm tip deformation transformation matrix differentiated with respect to nodal deflection displacement and the nodal slope displacement respectively.

These expressions are coded and solved for in the computer program listed in Appendix B.

APPENDIX B

SIMULATION PROGRAM

```
TITLE    THESI9V(2.11kg)
CONST    GKV=2.0,  GKP=1.0 ,DEA=1.0
* THESIS COPY
*
*         SIMULATION FOR DESIGN A CONTROL ALGORITHM
*         -----------------------------------------
*
*
*         THIS PROGRAM IS  TO CALCULATE THE REQUIRED TORQUE AND REQUIRED
*         CURRENT  TO CONTROL THE END POSITION OF THE SINGLE- LINK ARM.
*         THE ARM PARAMETERS ARE INPUTTED AND THE HYDRAULIC ACTUATION
*         DYNAMICS ARE INCLUDED IN THE SIMULATION.
*         THE INPUTS OF THE SYSTEM ARE THE DESIRED ANGLE OF THE END OF ARM.
*         THE OUTPUTS OF THE SYSTEM ARE THE TOTAL ANGLE, TOTAL ANGULAR
*         VELOCITY, ANGULAR ACCELERATION WHICH ARE FED BACK TO THE
*         CONTROLLER. THE CODING CONSISTS OF A MAIN
*         PROGRAM AND  SUBROUTINES AND ARE DESCRIBED BELOW.
*
*         THE FOLLOWING PARAMETERS ARE DEFINED:
*         1.A-EFFECTIVE CROSS-SECTIONAL AREA OF FLEXIBLE ARM
*         2.ARRDD-3X3 SECOND TIME DERIVATIVE OF ROTOR RESIDUAL ACCELERATION
*           MATRIX
*         3.ARTH-3X3 ROTOR TRANSFORMATION MATRIX DIFFERENTIATED WITH RESPECT
*           TO THETA
*         4.BE-EFFECTIVE BULK MODULUS OF FLUID
*         5.BIGF-2X1 RIGHT-HAND SIDE VECTOR FOR LARGE AND SMALL MOTION
*           ACCELERATIONS
*         6.BIGM-2X2 MATRIX OF LARGE AND SMALL MOTION ACCELERATION
*           COEFFICIENTS
*         7.CTM-TOTAL LEAKAGE COEFFICIENT OF THE ACTUATOR
*         8.DEFM-DISPLACEMENT DEFORMATION VARIABLE
*         9.DEFMD-TIME DERIVATIVE OF DISPLACEMENT DEFORMATION VARIABLE
*         10.DIFF,QERR,QERR1,FACTOR-DUMMY VARIABLES
*         11.DL1-3X3 DEFORMATION MATRIX
*         12.DL11-3X3 DEFORMATION MATRIX DIFFERENTIATED WITH RESPECT TO THE
*            DISPLACEMENT DEFORMATION VARIABLE
*         13.DL1D-3X3 FIRST TIME DERIVATIVE OF DEFORMATION MATRIX
*         14.DM-ACTUATOR DISPLACEMENT
*         15.E-MODULUS OF ELASTICITY OF STEEL
*         16.FN- RIGHT-HAND SIDE FOR SMALL MOTION ACCELERATIONS
*         17.FQ-RIGHT-HAND SIDE FOR LARGE MOTION ACCELERATIONS
*         18.G-3X1 GRAVITATIONAL ACCELERATION VECTOR
*         19.H11-1X3 LINK FIRST MOMENT OF INERTIA VECTOR
*         20.H21-1X3 LINK SHAPE MATRIX FIRST MOMENT OF INERTIAVECTOR
*         21.H41-1X3 LOAD FIRST MOMENT OF INERTIA VECTOR
*         22.KCE-TOTAL FLOW PRESSURE COEFFICIENT
*         23.PL-LOAD HYDRAULIC PRESSURE DROP
*         24.PS-HYDRAULIC SUPPLY PRESSURE
*         25.QL-FLOW DELIVERED FROM THE SERVOVALVE
*         26.SOL-2X1 VECTOR OF LARGE AND SMALL MOTION ACCELERATIONS
*         27.TE-TORQUE EFFICIENCY
*         28.TH-LARGE MOTION POSITION VARIABLE
*         29.THD-TIME DERIVATIVE OF LARGE MOTION VARIABLE
*         30.TORQUE-APPLIED TORQUE BY ACTUATOR
*         31.U-2X1 ARM TIP DEFORMATION OF  DISPLACEMENT
*         32.UD-ARM TIP DEFORMATION DIFFERENTIATED WITH RESPECT TO TIME
*         33.VT-TOTAL COMPRESSED VOLUME INCLUDING ACTUATOR LINES AND CHAMBERS
*         34.W-3X3 LINK TRANSFORMATION MATRIX
*         35.WD-3X3 FIRST TIME DERIVATIVE OF LINK TRANSFORMATION MATRIX
*         36.WRDD-3X3 SECOND TIME DERIVATIVE OF LINK RESIDUAL ACCELERATION
*            MATRIX
```

```
 *      37.WTH-3X3 TRANSFORMATION MATRIX DIFFERENTIATED WITH RESPECT TO
 *         THETA
 *      38.XIFRAC-VARIABLE FRACTIONAL AMOUNT OF INPUT CURRENT TO SERVO-
 *         VALVE
 *      39.XIINP-CURRENT INPUT EQUAL TO INITIAL AND FRACTIONAL AMOUNTS
 *      40.XIL-3X3 INERTIA MATRIX OF THE LOAD
 *      41.XIO-INITIAL INPUT CURRENT TO SERVOVALVE
 *      42.XIR-3X3 ROTOR INERTIA MATRIX
 *      43.XK11- PARTIAL LINK STIFFNESS
 *      44.XKN- LINK STIFFNESS
 *      45.XKV-SERVOVALVE SIZING CONSTANT
 *      46.XLL-LENGTH OF FLEXIBLE ARM
 *      47.XML-MASS OF LOAD
 *      48.XMNN-COEFFICIENT OF SMALL MOTION ACCELERATIONS IN THE
 *         SMALL MOTION DYNAMIC EQUATIONS
 *      49.XMNQ- COEFFICIENT OF LARGE MOTION ACCELERATIONS IN THE
 *         SMALL MOTION DYNAMIC EQUATIONS
 *      50.XMQN- COEFFICIENT OF SMALL MOTION ACCELERATIONS IN THE
 *         LARGE MOTION DYNAMICS EQUATION
 *      51.XMQQ-COEFFICIENT OF LARGE MOTION ACCELERATION IN THE LARGE MOTION
 *         DYNAMICS EQUATION
 *      52.XMQQP- DUMMY MATRIX FOR USE IN FORMULATING THE EQUATIONS OF
 *         MOTION
 *      53.XMR-MASS OF ACTUATOR ROTOR
 *      54.XMU-MASS DENSITY OF STEEL FLEXIBLE ARM
 *      55.XMX-FIRST MOMENT OF LOAD WITH RESPECT TO THE LOCAL COORDINATE
 *         Y AXIS
 *      56.ZI-AREA MOMENT OF INERTIA OF FLEXIBLE ARM
 *      57.TAg- TOTAL total angle
 *      58.TAG1- TOTAL ANGULAR VELOCITY
 *      59.TAG2- TOTAL ANGULAR ACCELERATION
 *      60.RETO- REQUIRED TORQUE
 *      61.REIC -REQUIRED CURRENT
 *
INITIAL
 *
 *         INITIAL VALUES OF PARAMETERS ARE INPUTTED VIA XINIT SUBROUTINE
 *
D      DIMENSION U( 1),XMQQ(1),XMQQP(1 ),DL1(3,3),WTH(3,3),ARTH(3,3),
D      #XIR(3,3),XMQN(1 ),UD(1 ),H11(1,3),G(3,1),H21(1,3),
D      #WRDD(3,3),DL1D(3,3),WD(3,3),ARRDD(3,3),H41(1,3),XK11(1 ),
D      #XMNQ(1 ),W(3,3),XMNN(1 ),XKN(1 ),FN(1 ),BIGM(2,2),
D      #BIGF(2,1),XIL(3,3),DL11(3,3),DEFMD(1),SOL(2),THD(1),
D      #A(1),E(1),ZI(1), FQ(1),GPOS(3),XITH(1),
D      #XMU(1),XLL(1),XML(1),XMR(1),XMX(1),TH(1),TORQUE(1),DEFM(1),
D      #PS(1),CTM(1),VT(1),BE(1),DM(1),XKV(1),TE(1),
D      #QL(1),PL(1),DIFF(1),XIINP(1),QERR1(1),QERR(1),FACTOR(1),
D      #rEIC(1), rETO(1),FTT(1),CTB(1),FCO(1),
D      #DEQ(1),DI(1),DEPL(1)
FIXED I
NOSORT
D      COMMON/FCDATA/C1,C2,  A1P,A2P,BETA1,BETA2

       C1=0.515462194
       C2=-0.205906794
       A1P=1.362220557
       A2P=0.981867539
       BETA1=1.877920950
       BETA2=4.701142847
       CALL XINIT(TH,THD,DEFM,DEFMD ,VO,POSO,A,XML,XMU,...
       XLL,XMR,E,ZI,PS ,CTM,VT,BE,DM,XKV,TE,QL,...
       PL,PLIC, REIC)
 *
DERIVATIVE
 *
 *         COEFFICIENTS FOR BOTH LARGE AND SMALL MOTION ACCELERATIONS
 *         AND THE RIGHT-HAND SIDES ARE COMPUTED IN THE FOLLOWING
 *         SUBROUTINES. ALSO,THE HYDRAULIC DYNAMICS ARE INCLUDED
 *         IN THE MAIN PROGRAM.
```

```
*
NOSORT
*
*          HYDRAULIC DYNAMICS
*
        XIINP(1)=DEIC(1)
        IF(PL(1).GT.PS(1)) GO TO 2
        GO TO 3
  2     PL(1)=PS(1)
  3     QERR1(1)=(XIINP(1)*XKV(1)*DSQRT(PS(1)-PL(1)))-(DM(1)*THD(1))
        QERR(1)=QERR1(1)/CTM(1)
        DIFF(1)=QERR(1)-PL(1)
        FACTOR(1)=VT(1)/(4.0D0*BE(1)*CTM(1))
        DIFF1(1)=DIFF(1)
SORT
        PL1=INTGRL(PLIC,DIFF1,1)
NOSORT
        PL(1)=PL1(1)/FACTOR(1)
        TORQUE(1)=TE(1)*PL(1)*DM(1)
*
*       MATRIX AND VECTOR FORMULATION SUBROUTINE
*
        CALL FORM(W,WTH,WD,DL1,DL1D,XIL,XIR,ARTH,WRDD,ARRDD,U,UD,...
        XMQQP,G,H11,H21,DL11 ,H41,XK11,A,XMU,XML,XLL,TH,THD,...
        DEFM,DEFMD ,E,ZI,XMR,XMX   )
*
*       COEFFICIENT OF LARGE MOTION ACCELERATION IN LARGE MOTION DYNAMICS
*       EQUATION SUBROUTINE
*
        CALL XLMMQQ(XMQQ,U,XMQQP,DL1,WTH,ARTH,XIL,XIR,A,XMU,TH,DEFM      )
*
*       COEFFICIENTS OF SMALL MOTION ACCELERATIONS IN LARGE MOTION DYNAMICS
*       EQUATION SUBROUTINE
*
        CALL XLMMQN(XMQN,A,XMU,XML,XLL,XMX,DEFM)
*
*       RIGHT-HAND SIDE FOR LARGE MOTION DYNAMICS EQUATION SUBROUTINE
*
        CALL XLMFQ(FQ,U,XMQQP,DL1,WTH,ARTH,XIL,XIR,UD,H11,G,H21,WRDD,...
        DL1D,WD,ARRDD,H41,TH,THD,DEFM,DEFMD ,A,XMU,XML,XLL,...
        TORQUE,FTT)
*
*       LINK STIFFNESS MATRIX SUBROUTINE
*
        CALL SMKN(XKN,XK11,XMQQP,A,XMU,THD)
*
*       COEFFICIENTS OF LARGE MOTION ACCELERATION IN SMALL MOTION
*       DYNAMICS EQUATIONS SUBROUTINE
*
        CALL SMMNQ(XMNQ,DL1,WTH,XIL,DL11        ,W,TH,DEFM,       A,XMU,...
        XLL)
*
*       RIGHT-HAND SIDE OF SMALL MOTION DYNAMICS EQUATIONS SUBROUTINE
*
        CALL SMFN(FN,H21,W,G,WRDD,DL1,XIL,DL11,       WD,DL1D,H41,TH,...
        THD,DEFM,DEFMD)
*
*       COEFFICIENTS OF SMALL MOTION ACCELERATIONS IN SMALL MOTION DYNAMICS
*       EQUATIONS SUBROUTINE
*
        CALL SMMNN(XMNN,XMQQP,XML,A,XMU)
*
*       ACCELERATION COEFFICIENTS MATRIX AND RIGHT-HAND SIDE VECTOR
*       FORMULATION SUBROUTINE
*
        CALL BIGFOR(BIGM,BIGF,XMQQ,XMQN,FQ,XMNQ,XMNN,XKN,FN,U,DEFMD)
*
*       LINEAR EQUATION SOLVER FOR ACCELERATIONS SUBROUTINE
*
        CALL XLEQ(BIGM,BIGF,SOL)
```

73

```
*
*
*       INTEGRATE ACCELERATIONS AND THEN VELOCITY TO GET LARGE MOTION
*       ANGULAR POSITION AND SMALL MOTION,LOCAL COORDINATE,TIP POSITION
*
        DO 5 I=1,2
        SOL1(I)=SOL(I)
    5   CONTINUE
SORT
        VEL=INTGRL(VO,SOL1,2)
NOSORT
        THD(1)=VEL(1)
        DEFMD(1)=VEL(2)
        DO 10 I=1,2
        VEL1(I)=VEL(I)
   10   CONTINUE
SORT
        POS=INTGRL(POSO,VEL1,2)
NOSORT
        TH(1)=POS(1)
             ANG=TH(1)
                  CUR=XIINP(1)
        DEFM(1)=POS(2)
        AC1=SOL1(1)
        AC2=SOL1(2)
        VE1=VEL(1)
        VE2=VEL(2)
        PO1=POS(1)
        PO2=POS(2)
*
*       ANGULAR ACCELERATION,VELOCITY,ANGLE.
*
        TAG2=AC1+AC2/0.9985D0
        TAG1=VE1+VE2/0.9985D0
        TAG =PO1  +PO2/0.9985D0
*
*       COEFFICIENT OF THE CONTROL ALGORITHM
        CALL CCO(XMQQ,XMQN,XMNN,XMNQ,FTT,FN,XKN,...
        DEFM,CTB,FCO,    DEFMD)
*
*       CALCULATE THE REQUIRED TORQUE
*
        CALL REQTO(RETO,CTB,TAG1,TAG,FCO,GKV,GKP,DEA)
*
*       INVERSE HYDRAUIC  DYNAMIC
*
*       CALCULATE THE REQUIRED CURRENT
*
        CALL REC(RETO,PL,TE,DM,VT,THD,PS,CTM,XKV,REIC,BE)
        TORK=DETO(1)
        CU=DEIC(1)
TERMINAL
METHOD ADAMS
PRINT 12.0E-3, TAG ,TORK ,CU
SAVE 6.0E-3,TAG
CONTROL FINTIM=5.000 ,DELT=0.006
END
PARAM GKV=4.0 ,GKP=4.0
END
PARAM   GKV=8.0,GKP=16.0
GRAPH (G1,DE=TEK618) TIME(UN=SEC),TORK
LABEL (G1) LOAD=2.1151 KG
GRAPH (G2,DE=TEK618) TIME(UN=SEC),TAG
LABEL (G2) LOAD=2.1151 KG
GRAPH (G3,DE=TEK618)TIME,CU
LABEL (G3) LOAD=2.1151 KG
GRAPH (G4,DE=TEK618) TIME,TAG1
LABEL (G4) LOAD=2.1151 KG
GRAPH (G5,DE=TEK618) TIME,TAG2
LABEL (G5) LOAD=2.1151 KG
```

```fortran
      END
      STOP
*
*     LISTING OF SUBROUTINES
*
      FORTRAN
*
*.........................................................
*
        SUBROUTINE XINIT(TH,THD,DEFM,DEFMD ,VO,POSO,A,ML,MU,LL,
     #MR,E,ZI,PS,CTM,VT,BE,DM,KV,TE,QL,PL,PLIC ,REIC   )
        REAL*8 VO(2),POSO(2),ML,MU,LL,MR,TH,THD,DEFM,DEFMD,
     #A,E,ZI,ITORQ,PS,  CTM,VT,BE,DM,KV,TE,QL,PL,PLIC,
     #C1,C2,AP1,AP2,BETA1 ,BETA2,rEIC ,
     #DEPL,DEQ,DI,DEFP,DEIC1,DEIC2
*
        ITORQ= 44.5549505100000000
        DM=6.2271D-05
        TE=.90000000000000
        PL=ITORQ/(DM*TE)
        PLIC=PL
        CTM=3.7064772D-13
        QL=CTM*PL
        KV=2.402963D-09
        PS=1.37888D+07
        VT=3.05127D-04
        BE=690.D6
        A=6.17795D-04
        ML=2.1151000000000
        MU=7861.05000000000000
        LL=0.99850000000000
        MR=9.00011451000000
        E=2.0D11
        ZI=4.065D-10
        VO(1)=0.000000000000000
        VO(2)=0.000000000000000
        POSO(1)=0.00000000000000
        POSO(2)=-.14606414900000
        TH=POSO(1)
        THD=VO(1)
        DEFM=POSO(2)
        DEFMD=VO(2)
        REIC=4.0
        RETURN
        END
*
*.........................................................
        DOUBLE PRECISION FUNCTION ONE(X)
        REAL*8 C1,C2,      A1P,A2P,BETA1,BETA2
        COMMON/FCDATA/C1,C2      ,A1P,A2P,BETA1,BETA2
        ONE=                          C1*(A1P*(COS(BETA1*X)+COSH
     #                    (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                    C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                    (SIN(BETA2*X)+SINH(BETA2*X)))
        RETURN
        END
C
        DOUBLE PRECISION FUNCTION TWO(X)
        REAL*8 C1,C2,      A1P,A2P,BETA1,BETA2
        COMMON/FCDATA/C1,C2,      A1P,A2P,BETA1,BETA2
        TWO=                          (C1*(A1P*(COS(BETA1*X)+COSH
     #                    (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                    C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                    (SIN(BETA2*X)+SINH(BETA2*X))))**2
        RETURN
        END
        DOUBLE PRECISION FUNCTION SIX(X)
        REAL*8 C1,C2,      A1P,A2P,BETA1,BETA2
        COMMON/FCDATA/C1,C2,      A1P,A2P,BETA1,BETA2
```

75

```fortran
      SIX=                               .9985*(C1*(A1P*(COS(BETA1*X)+COSH
     #                   (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                        C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                            (SIN(BETA2*X)+SINH(BETA2*X))))+
     #                          X*(C1*(A1P*(COS(BETA1*X)+COSH
     #                   (BETA1*X))+(SIN(BETA1*X)+SINH(BETA1*X)))+
     #                        C2*(A2P*(COS(BETA2*X)+COSH(BETA2*X))+
     #                            (SIN(BETA2*X)+SINH(BETA2*X))))
      RETURN
      END


      DOUBLE PRECISION FUNCTION EIGHT(X)
      REAL*8 C1,C2,        A1P,A2P,BETA1,BETA2
      COMMON/FCDATA/C1,C2,         A1P,A2P,BETA1,BETA2
      EIGHT=                             (C1*BETA1*BETA1*(A1P*(-COS
     #          (BETA1*X)+COSH(BETA1*X))+(-SIN(BETA1*X)+SINH(BETA1*X)))+
     #               C2*BETA2*BETA2*(A2P*(-COS(BETA2*X)+COSH(BETA2*X))+
     #                    (-SIN(BETA2*X)+SINH(BETA2*X))))**2
      RETURN
      END
*...............................................................................
*...............................................................................
*
      SUBROUTINE FORM(W,WTH,WD,DL1,DL1D,XIL,XIR,ARTH,WRDD,ARRDD,U,UD,
     #XMQQP,G,H11,H21,DL11       ,H41,XK11,A,MU,ML,LL,TH,THD,DEFM,DEFMD,
     #          E,ZI,MR,MX          )
      REAL*8 W(3,3),WTH(3,3),WD(3,3),DL1(3,3),DL1D(3,3),XIL(3,3)
      REAL*8 XIR(3,3),ARTH(3,3),WRDD(3,3),ARRDD(3,3),U        ,UD
      REAL*8 XMQQP       ,G(3,1),H11(1,3),H21(1,3),DL11(3,3)
      REAL*8 H41(1,3),XK11       ,MU,ML,LL,MR,MX,TH
      REAL*8 THD,DEFM,DEFMD                       ,A,E,ZI
      REAL*8 C1,A1P,BETA1
      REAL*8 ONE,TWO,EIGHT,Y
      EXTERNAL ONE,TWO,EIGHT
      W(1,1)=1.00000000000000
      W(1,2)=0.00000000000000
      W(1,3)=0.00000000000000
      W(2,1)=LL*DCOS(TH)
      W(2,2)=DCOS(TH)
      W(2,3)=-DSIN(TH)
      W(3,1)=LL*DSIN(TH)
      W(3,2)=DSIN(TH)
      W(3,3)=DCOS(TH)
      WTH(1,1)=0.00000000000000
      WTH(1,2)=0.00000000000000
      WTH(1,3)=0.00000000000000
      WTH(2,1)=-LL*DSIN(TH)
      WTH(2,2)=-DSIN(TH)
      WTH(2,3)=-DCOS(TH)
      WTH(3,1)=LL*DCOS(TH)
      WTH(3,2)=DCOS(TH)
      WTH(3,3)=-DSIN(TH)
      WD(1,1)=0.00000000000000
      WD(1,2)=0.00000000000000
      WD(1,3)=0.00000000000000
      WD(2,1)=-LL*DSIN(TH)*THD
      WD(2,2)=-DSIN(TH)*THD
      WD(2,3)=-DCOS(TH)*THD
      WD(3,1)=LL*DCOS(TH)*THD
      WD(3,2)=DCOS(TH)*THD
      WD(3,3)=-DSIN(TH)*THD
      DL1(1,1)=1.00000000000000
      DL1(1,2)=0.00000000000000
      DL1(1,3)=0.00000000000000
      DL1(2,1)=0.00000000000000
      DL1(2,2)=1.00000000000000
      DL1(2,3)=0.0000000000000
      DL1(3,1)=DEFM
```

76

```
DL1(3,2)=0.000000000000000
DL1(3,3)=1.00000000000000
DL1D(1,1)=0.00000000000000
DL1D(1,2)=0.00000000000000
DL1D(1,3)=0.00000000000000
DL1D(2,1)=0.00000000000000
DL1D(2,2)=0.000000000000
DL1D(2,3)=0.000000000000
DL1D(3,1)=DEFMD
DL1D(3,2)=0.000000000000
DL1D(3,3)=0.00000000000000
XIL(1,1)=ML
XIL(1,2)=0.01678545900000
XIL(1,3)=.00000000000000
XIL(2,1)=0.01678545900000
XIL(2,2)=1.77679D-04
XIL(2,3)=0.00000000000000
XIL(3,1)=.00000000000000
XIL(3,2)=0.00000000000000
XIL(3,3)=2.986791D-03
MX=.01678545900000
XIR(1,1)=MR
XIR(1,2)=0.000000000000000
XIR(1,3)=0.000000000000000
XIR(2,1)=0.000000000000000
XIR(2,2)=.027467130000000
XIR(2,3)=0.000000000000000
XIR(3,1)=0.000000000000000
XIR(3,2)=0.000000000000000
XIR(3,3)=.02746713000000
ARTH(1,1)=0.00000000000000
ARTH(1,2)=0.00000000000000
ARTH(1,3)=0.00000000000000
ARTH(2,1)=0.00000000000000
ARTH(2,2)=-DSIN(TH)
ARTH(2,3)=-DCOS(TH)
ARTH(3,1)=0.00000000000000
ARTH(3,2)=DCOS(TH)
ARTH(3,3)=-DSIN(TH)
WRDD(1,1)=0.00000000000000
WRDD(1,2)=0.00000000000000
WRDD(1,3)=0.00000000000000
WRDD(2,1)=-LL*DCOS(TH)*(THD**2)
WRDD(2,2)=-DCOS(TH)*(THD**2)
WRDD(2,3)=DSIN(TH)*(THD**2)
WRDD(3,1)=-LL*DSIN(TH)*(THD**2)
WRDD(3,2)=-DSIN(TH)*(THD**2)
WRDD(3,3)=-DCOS(TH)*(THD**2)
ARRDD(1,1)=0.00000000000000
ARRDD(1,2)=0.00000000000000
ARRDD(1,3)=0.00000000000000
ARRDD(2,1)=0.00000000000000
ARRDD(2,2)=-DCOS(TH)*(THD**2)
ARRDD(2,3)=DSIN(TH)*(THD**2)
ARRDD(3,1)=0.00000000000000
ARRDD(3,2)=-DSIN(TH)*(THD**2)
ARRDD(3,3)=-DCOS(TH)*(THD**2)
U      =DEFM
UD     =DEFMD
CALL DQG4(-LL,0.D0,TWO,Y)
XMQQP     =Y
G(1,1)=0.00000000000000
G(2,1)=0.0000000000000
G(3,1)=-9.80660000000000
H11(1,1)=4.85651900000000
H11(1,2)=-2.42825869300000
H11(1,3)=0.00000000000000
H21(1,1)=0.00000000000000
H21(1,2)=0.00000000000000
CALL DQG4(-LL,-0.D0,ONE ,Y)
```

77

```
          H21(1,3)=A*MU*Y
          DO 50 I=1,3
          DO 60 J=1,3
          DL11(I,J)=0.00000000000000
   60     CONTINUE
   50     CONTINUE
          DL11(3,1)=1.00000000000000
          H41(1,1)=ML
          H41(1,2)=0.000000000000000
          H41(1,3)=.016785459000000000
          CALL DQG4(-LL,0.D0,EIGHT,Y)
          XK11      =Y*E*ZI
          RETURN
          END
*
*..........................................................................
*
          SUBROUTINE XLMMQQ(MQQ,U,XMQQP,DL1,WTH,ARTH,XIL,XIR,A,MU,TH,DEFM)
          REAL*8 MQQ,UT,P,DL1T(3,3),WTHT(3,3),ARTHT(3,3),P1(3,3)
          REAL*8 P2(3,3),P3(3,3),P4(3,3),P5(3,3),P6(3,3),P7(3,3),MU
          REAL*8 U ,XMQQP,DL1(3,3),WTH(3,3),ARTH(3,3),XIL(3,3)
          REAL*8 XIR(3,3),A,TH,DEFM ,SP,TP
          M=2
          L=1
          N=3
          MQQ=0.00000000000000
          CALL TRANS(U,UT,L,L)
          CALL MATMUL(UT,XMQQP,L,L,L,P)
          CALL MATMUL(P,U,L,L,L,SP)
          CALL TRANS(DL1,DL1T,N,N)
          CALL TRANS(ARTH,ARTHT,N,N)
          CALL TRANS(WTH,WTHT,N,N)
          CALL MATMUL(WTH,DL1,N,N,N,P1)
          CALL MATMUL(P1,XIL,N,N,N,P2)
          CALL MATMUL(P2,DL1T,N,N,N,P3)
          CALL MATMUL(P3,WTHT,N,N,N,P4)
          CALL MATMUL(ARTH,XIR,N,N,N,P5)
          CALL MATMUL(P5,ARTHT,N,N,N,P6)
          CALL MATADD(P4,P6,N,N,P7)
          CALL TRACE(P7,N,TP)
          MQQ=((1./3.)*A*MU) + (A*MU*SP) + TP
          RETURN
          END
*
*..........................................................................
*
          SUBROUTINE XLMMQN(XMQN,A,MU,ML,LL,MX       ,DEFM              )
          REAL*8 XMQN       ,MU,ML,LL,MX,A,        DEFM
          REAL*8 C1,A1P,BETA1,Y,SIX ,C2,A2P,BETA2
          EXTERNAL SIX
          CALL DQG4(-LL,0.D0,SIX ,Y)
          XMQN      =(A*MU*Y)+(ML*LL)+MX
          RETURN
          END
*
*..........................................................................
*
          SUBROUTINE XLMFQ(FQ,U,XMQQP,DL1,WTH,ARTH,XIL,XIR,UD,H11,G,H21,
         #WRDD,DL1D,WD,ARRDD,H41,TH,THD,DEFM,DEFMD ,A,MU,ML,LL,
         #TORQUE,FTT)
          REAL*8 FQP,P ,P1(1,3),P2(1,3),P3(1,3),P4(3,3),P5(3,3)
          REAL*8 P6(3,3),P7(3,3),P8(3,3),P9(3,3),P10(3,3),P11(1,3),P12(1,3)
          REAL*8 FPFH(3,3),FHP(3,3),FPT(3,3),DL1DT(3,3),WDT(3,3),ARRDDT(3,3)
          REAL*8 UT,DL1T(3,3),WRDDT(3,3),FPF(3,3),FPS(3,3),WTHT(3,3)
          REAL*8 U,XMQQP ,DL1(3,3),WTH(3,3),ARTH(3,3),XIL(3,3)
          REAL*8 XIR(3,3),UD,H11(1,3),G(3,1),H21(1,3),WRDD(3,3)
          REAL*8 DL1D(3,3),WD(3,3),ARRDD(3,3),H41(1,3),MU,LL,ML
          REAL*8 A,TORQUE,FQ,TH,THD,DEFM,DEFMD
          REAL*8 FP,SP,TP,TFP,FTHP ,FTT
          M=2
```

78

```fortran
          L=1
          N=3
          CALL TRANS(U,UT,L,L)
          FQP=XMQQP *THD
          CALL MATMUL(UT,FQP,L,L,L,P)
          CALL MATMUL(P,UD,L,L,L,FP)
          CALL TRANS(WTH,WTHT,N,N)
          CALL MATMUL(H11,WTHT,L,N,N,P1)
          CALL MATMUL(P1,G,L,N,L,SP)
          CALL MATMUL(UT,H21,L,M,N,P2)
          CALL MATMUL(P2,WTHT,L,N,N,P3)
          CALL MATMUL(P3,G,L,N,L,TP)
          CALL TRANS(DL1,DL1T,N,N)
          CALL TRANS(WRDD,WRDDT,N,N)
          CALL MATMUL(WTH,DL1,N,N,N,P4)
          CALL MATMUL(P4,XIL,N,N,N,P5)
          CALL MATMUL(P5,DL1T,N,N,N,P6)
          CALL MATMUL(P6,WRDDT,N,N,N,FPF)
          CALL TRANS(DL1D,DL1DT,N,N)
          CALL TRANS(WD,WDT,N,N)
          CALL MATMUL(WTH,DL1,N,N,N,P7)
          CALL MATMUL(P7,XIL,N,N,N,P8)
          CALL MATMUL(P8,DL1DT,N,N,N,P9)
          CALL MATMUL(P9,WDT,N,N,N,FPS)
          CALL TRANS(ARRDD,ARRDDT,N,N)
          CALL MATMUL(ARTH,XIR,N,N,N,P10)
          CALL MATMUL(P10,ARRDDT,N,N,N,FPT)
          DO 30 I=1,3
          DO 40 J=1,3
          FPS(I,J)=FPS(I,J)*2.
   40     CONTINUE
   30     CONTINUE
          CALL MATADD(FPF,FPS,N,N,FPFH)
          CALL MATADD(FPFH,FPT,N,N,FHP)
          CALL TRACE(FHP,N,TFP)
          CALL MATMUL(H41,DL1T,L,N,N,P11)
          CALL MATMUL(P11,WTHT,L,N,N,P12)
          CALL MATMUL(P12,G,L,N,L,FTHP)
          FTT=(-2.*A*MU*FP) + SP + TP - TFP + FTHP
          FQ=FTT+TORQUE
          RETURN
          END
*
*.........................................................
*
          SUBROUTINE SMKN(XKN,XK11,XMQQP,A,MU,THD)
          REAL*8 XKN,KNP ,XMQQP ,XK11 ,A,THD,MU
          KNP =XMQQP *(-A)*MU*(THD**2)
          XKN =KNP  +XK11
          RETURN
          END
*
*.........................................................
*
          SUBROUTINE SMMNQ(XMNQ,DL1,WTH,XIL,DL11,W,TH,DEFM,A,MU,
         #LL)
          REAL*8 XMNQ ,DL11T(3,3),WT(3,3),P1(3,3),P2(3,3)
          REAL*8 P3(3,3),P4(3,3),DL1(3,3),WTH(3,3),XIL(3,3)
          REAL*8 W(3,3),DL11(3,3) ,TH,DEFM ,A,MU,LL
          REAL*8 C1,A1P,BETA1
          REAL*8 SIX,Y ,C2,A2P,BETA2
          EXTERNAL SIX
          M=2
          L=1
          N=3
          CALL TRANS(DL11,DL11T,N,N)
          CALL TRANS(W,WT,N,N)
          CALL MATMUL(WTH,DL1,N,N,N,P1)
          CALL MATMUL(P1,XIL,N,N,N,P2)
          CALL MATMUL(P2,DL11T,N,N,N,P3)
```

79

```
            CALL MATMUL(P3,WT,N,N,N,P4)
            CALL TRACE(P4,N,TFP1)
            CALL DQG4(-LL,0.D0,SIX ,Y)
            XMNQ =TFP1 + (Y*A*MU)
            RETURN
            END
*
*.............................................................................
*
            SUBROUTINE SMFN(FN,H21,W,G,WRDD,DL1,XIL,DL11,        WD,DL1D,H41,TH,
        #THD,DEFM,DEFMD)
            REAL*8 FN,P1(1,3),P2(3,3),P3(3,3),P4(3,3),P5(3,3),P6(3,3)
            REAL*8 P7(3,3),P8(3,3),P9(3,3)
            REAL*8 P14(1,3),P15(1,3) ,TP,FP ,SP
            REAL*8 FN1(3,3), G(3,1),H21(1,3),WRDD(3,3),DL1D(3,3)
            REAL*8 WD(3,3),H41(1,3),XIL(3,3),W(3,3),DL11(3,3)
            REAL*8 DL1(3,3),DL11T(3,3),WT(3,3)
            REAL*8 TH,THD,DEFM,DEFMD,TFN1,FN3
            M=2
            L=1
            N=3
            CALL TRANS(W,WT,N,N)
            CALL MATMUL(H21,WT,L,N,N,P1)
            CALL MATMUL(P1,G,L,N,L,FP)
            CALL TRANS(DL11,DL11T,N,N)
            CALL MATMUL(WRDD,DL1,N,N,N,P2)
            CALL MATMUL(P2,XIL,N,N,N,P3)
            CALL MATMUL(P3,DL11T,N,N,N,P4)
            CALL MATMUL(P4,WT,N,N,N,P5)
            CALL MATMUL(WD,DL1D,N,N,N,P6)
            CALL MATMUL(P6,XIL,N,N,N,P7)
            CALL MATMUL(P7,DL11T,N,N,N,P8)
            CALL MATMUL(P8,WT,N,N,N,P9)
            DO 10 I=1,3
            DO 20 J=1,3
            P9(I,J)= P9(I,J)*2.
    20     CONTINUE
    10     CONTINUE
            CALL MATADD(P5,P9,N,N,FN1)
            CALL TRACE(FN1,N,TFN1)
            SP =TFN1
            CALL MATMUL(H41,DL11T,L,N,N,P14)
            CALL MATMUL(P14,WT,L,N,N,P15)
            CALL MATMUL(P15,G,L,N,L,FN3)
            TP =FN3
            FN=FP -SP + TP
            RETURN
            END
*
*.............................................................................
*
            SUBROUTINE SMMNN(XMNN,XMQQP,ML,A,MU                    )
            REAL*8 XMNN,XMQQP ,ML,MU,A
            XMNN =ML
            XMNN =XMNN + XMQQP *A*MU
            RETURN
            END
*
*.............................................................................
*
            SUBROUTINE MATMUL(A,B,M,L,N,C)
            REAL*8 A(M,L),B(L,N),C(M,N)
            DO 10 I=1,M
            DO 20 J=1,N
            C(I,J)=0.0
            DO 30 INDEX=1,L
            C(I,J)=C(I,J) + A(I,INDEX)*B(INDEX,J)
    30     CONTINUE
    20     CONTINUE
    10     CONTINUE
```

80

```
        RETURN
        END
*
*..........................................................
*
        SUBROUTINE TRANS(A,B,M,L)
        REAL*8 A(M,L),B(L,M)
        DO 10 I=1,M
        DO 20 J=1,L
        B(J,I)=A(I,J)
  20    CONTINUE
  10    CCNTINUE
        RETURN
        END
*
*..........................................................
*
        SUBROUTINE TRACE(A,M,TRAC)
        REAL*8 A(M,M)
        TRAC=0.0
        DO 10 I=1,M
        TRAC=TRAC + A(I,I)
  10    CONTINUE
        RETURN
        END
*
*       MATRIX ADDITION SUBROUTINE
*
        SUBROUTINE MATADD(A,B,M,L,C)
        REAL*8 A(M,L),B(M,L),C(M,L)
        DO 10 I=1,M
        DO 20 J=1,L
        C(I,J)=A(I,J) + B(I,J)
  20    CONTINUE
  10    CONTINUE
        RETURN
        END
*
*..........................................................
*
        SUBROUTINE BIGFOR(BIGM,BIGF,MQQ,XMQN,FQ,XMNQ,XMNN,XKN,FN,U,DEFMD)
        REAL*8 BIGM(2,2),BIGF(2,1),XMQN       ,XMNQ       ,XMNN       ,XKN
        REAL*8 FN       ,U       ,MQQ,P       ,FQ,DEFMD
        M=2
        L=1
        BIGM(1,1)=MQQ
        BIGM(1,2)=XMQN
        BIGM(2,1)=XMNQ
        BIGM(2,2)=XMNN
        BIGF(1,1)=FQ
        CALL MATMUL(XKN,U,L,L,L,P)
        BIGF(2,1)=FN       -P
        RETURN
        END
*
*..........................................................
*
        SUBROUTINE XLEQ(BIGM,BIGF,SOL)
        REAL*8 BIGM(2,2),BIGF(2,1),SOL(2),WKAREA(18)
        M=1
        N=2
        CALL LEQT2F (BIGM,M,N,N,BIGF,M,WKAREA,IER)
        DO 10 I=1,2
        SOL(I)=BIGF(I,1)
  10    CONTINUE
        RETURN
        END
*
*
*    SUBROUTINE LEQT2F (IMSL)
```

81

```
*
*    PURPOSE                    - LINEAR EQUATION SOLUTION - FULL STORAGE
*                                MODE - HIGH ACCURACY SOLUTION
*
*    USAGE                      - CALL LEQT2F (A,M,N,IA,B,IDGT,WKAREA,IER)
*
*    ARGUMENTS    A             - INPUT MATRIX OF DIMENSION N BY N CONTAINING
*                                  THE COEFFICIENT MATRIX OF THE EQUATION
*                                  AX = B.
*                 M             - NUMBER OF RIGHT-HAND SIDES. (INPUT)
*                 N             - ORDER OF A AND NUMBER OF ROWS IN B. (INPUT)
*                 IA            - ROW DIMENSION OF A AND B EXACTLY AS SPECIFIED
*                                  IN THE DIMENSION STATEMENT IN THE CALLING
*                                  PROGRAM. (INPUT)
*                 B             - INPUT MATRIX OF DIMENSION N BY M CONTAINING
*                                  THE RIGHT-HAND SIDES OF THE EQUATION AX = B.
*                                  ON OUTPUT, THE N BY M MATRIX OF SOLUTIONS
*                                  REPLACES B.
*                 IDGT          - INPUT OPTION.
*                                  IF IDGT IS GREATER THAN 0, THE ELEMENTS OF
*                                  A AND B ARE ASSUMED TO BE CORRECT TO IDGT
*                                  DECIMAL DIGITS AND THE ROUTINE PERFORMS
*                                  AN ACCURACY TEST.
*                                  IF IDGT EQUALS 0, THE ACCURACY TEST IS
*                                  BYPASSED.
*                                  ON OUTPUT, IDGT CONTAINS THE APPROXIMATE
*                                  NUMBER OF DIGITS IN THE ANSWER WHICH
*                                  WERE UNCHANGED AFTER IMPROVEMENT.
*                 WKAREA        - WORK AREA OF DIMENSION GREATER THAN OR EQUAL
*                                  TO N**2+3N.
*                 IER           - ERROR PARAMETER. (OUTPUT)
*                                  WARNING ERROR
*                                    IER = 34 INDICATES THAT THE ACCURACY TEST
*                                      FAILED. THE COMPUTED SOLUTION MAY BE IN
*                                      ERROR BY MORE THAN CAN BE ACCOUNTED FOR
*                                      BY THE UNCERTAINTY OF THE DATA. THIS
*                                      WARNING CAN BE PRODUCED ONLY IF IDGT IS
*                                      GREATER THAN 0 ON INPUT. (SEE THE
*                                      CHAPTER L PRELUDE FOR FURTHER DISCUSSION.)
*                                  TERMINAL ERROR
*                                    IER = 129 INDICATES THAT THE MATRIX IS
*                                      ALGORITHMICALLY SINGULAR. (SEE THE
*                                      CHAPTER L PRELUDE).
*                                    IER = 131 INDICATES THAT THE MATRIX IS TOO
*                                      ILL-CONDITIONED FOR ITERATIVE IMPROVEMENT
*                                      TO BE EFFECTIVE.
*    REQD. IMSL ROUTINES - SINGLE/LUDATN,LUELMN,LUREFN,UERTST,UGETIO
*                        - DOUBLE/LUDATN,LUELMN,LUREFN,UERTST,UGETIO,
*                                 VXADD,VXMUL,VXSTO
*.............................................................
      SUBROUTINE LEQT2F (A,M,N,IA,B,IDGT,WKAREA,IER)
*
      DIMENSION             A(IA,1),B(IA,1),WKAREA(1)
      DOUBLE PRECISION      A,B,WKAREA,D1,D2,WA
*                                FIRST EXECUTABLE STATEMENT
*                                INITIALIZE IER
      IER=0
      JER=0
      J = N*N+1
      K = J+N
      MM = K+N
      KK = 0
      MM1 = MM-1
      JJ=1
      DO 5 L=1,N
         DO 5 I=1,N
            WKAREA(JJ)=A(I,L)
            JJ=JJ+1
    5 CONTINUE
*                                DECOMPOSE A

                                82
```

```
          CALL LUDATN (WKAREA,N,N,A,IA,IDGT,D1,D2,WKAREA(J),WKAREA(K),
      *              WA,IER)
          IF (IER.GT.128) GO TO 25
          IF (IDGT .EQ. 0 .OR. IER .NE. 0) KK = 1
          DO 15 I = 1,M
*                                    PERFORMS THE ELIMINATION PART OF
*                                    AX = B
            CALL LUELMN (A,IA,N,B(1,I),WKAREA(J),WKAREA(MM))
*                                    REFINEMENT OF SOLUTION TO AX = B
            IF (KK .NE. 0)
      *     CALL LUREFN (WKAREA,N,N,A,IA,B(1,I),IDGT,WKAREA(J),WKAREA(MM),
      *              WKAREA(K),WKAREA(K),JER)
            DO 10 II=1,N
               B(II,I) = WKAREA(MM1+II)
   10      CONTINUE
            IF (JER.NE.0) GO TO 20
   15 CONTINUE
          GO TO 25
   20 IER = 131
   25 JJ=1
          DO 30 J = 1,N
            DO 30 I = 1,N
               A(I,J)=WKAREA(JJ)
               JJ=JJ+1
   30 CONTINUE
          IF (IER .EQ. 0) GO TO 9005
 9000 CONTINUE
          CALL UERTST (IER,6HLEQT2F)
 9005 RETURN
          END
*
*
*
*      ............................................................
*
*          SUBROUTINE DQG4 (IMSL SUBROUTINE)
*
*          PURPOSE
*             TO COMPUTE INTEGRAL(FCT(X), SUMMED OVER X FROM XL TO XU)
*
*          USAGE
*             CALL DQG4 (XL,XU,FCT,Y)
*             PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT
*          DESCRIPTION OF PARAMETERS
*             XL      - DOUBLE PRECISION LOWER BOUND OF THE INTERVAL.
*             XU      - DOUBLE PRECISION UPPER BOUND OF THE INTERVAL.
*             FCT     - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION
*                       SUBPROGRAM USED.
*             Y       - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.
*          METHOD
*             EVALUATION IS DONE BY MEANS OF 4-POINT GAUSS QUADRATURE
*             FORMULA, WHICH INTEGRATES POLYNOMIALS UP TO DEGREE 7
*             EXACTLY. FOR REFERENCE, SEE
*             V.I.KRYLOV, APPROXIMATE CALCULATION OF INTEGRALS,
*             MACMILLAN, NEW YORK/LONDON, 1962, PP.100-111 AND 337-340.
*
          SUBROUTINE DQG4(XL,XU,FCT,Y)
*
          DOUBLE PRECISION XL,XU,Y,A,B,C,FCT
          A=.5D0*(XU+XL)
          B=XU-XL
          C=.43056815579702629D0*B
          Y=.17392742256872693D0*(FCT(A+C)+FCT(A-C))
          C=.16999052179242813D0*B
          Y=B*(Y+.32607257743127307D0*(FCT(A+C)+FCT(A-C)))
          RETURN
          END
*
*..............................................................
          SUBROUTINE CCO(XMQQ,XMQN,XMNN,XMNQ,FTT,FN,XKN,DEFM,CTB,FCO,
      #DEFMD)
          REAL*8 XMQQ,XMNN,XMQN,FTT,FN,XKN,U,CTB,FCO,CTA,B11,B22
```

83

```
          REAL*8 B33,B44,F11,F22,DEFMD,F33
              B11=XMQQ/0.9985
              B22=XMQN-B11
              B33=XMNQ/0.9985
              B44=XMNN-B33
              CTA=B22/B44
              CTB=XMQQ-CTA*XMNQ
              F11=CTA*FN
              F22=(CTA*XKN)*DEFM
              F33=(CTA*0.00)*DEFMD
              FCO=F11-F22-FTT -F33
              RETURN
              END
*...........................................................
*
          SUBROUTINE REQTO(RETO,CTB,TAG1,TAG,FCO,GKV,GKP,DEA)
              REAL*8   DETO1,DETO2,rETO,GKV,GKP,DEA,TAG1,TAG,FCO,CTB
              DETO1=(-GKV*TAG1)+GKP*(DEA-TAG)
              DETO2=CTB*DETO1
              RETO=DETO2+FCO
              RETURN
              END
*...........................................................
*
          SUBROUTINE REC(RETO,PL,TE,DM,VT,THD,PS,CTM,XKV,DEIC,BE)
          REAL*8 RETO,PL, TE,DM,VT,THD,PS,CTM,XKV,REIC ,BE
          REAL*8 DI,DEQ,DEFP,DEIC1,DEIC2
          DEPL  =DETO   /(TE   *DM   )
          DI    =VT   *(DEPL   -PL   )/(4.0D0*BE   *0.006D0)
          DEQ   =DM   *THD   +CTM   *DEPL   +DI
          DEFP = (PS   - DEPL   )
          IF(DEFP .LT.0.0)  GO TO 13
          DEIC1=DSQRT(DEFP)
          GO TO 14
13        DEFP=-DEFP
          DEIC1=DSQRT(DEFP)
14        DEIC2=DEIC1*XKV
          REIC   =DEQ   /(DEIC2   )
          RETURN
          END
```

LIST OF REFERENCES

1. Whitney, D. E., Book, W. J., and Lynch, P. M,. "Design and Control Configurations for Industrial and Space Manipulators." Proc. JACC, 1974.

2. Book, W. J., "Feedback Control of Two Beam, Two Joint Systems With Distributed Flexibility," Journal of Dynamic Systems, Measurement and Control, Vol. 97, No. 4, pp. 424-430, Dec. 1975.

3. Sunada, H. H., and Dubowsky, S., "On the Dynamic Analysis and Behavior of Industrial Robotic Manipulators with Elastic Members," ASME Jounal of Mechanisms, Transmissions, and Automation in Design, Vol. 105, pp. 42-51, Mar. 1983.

4. Chang, L. W., *Dynamic Analysis of Robotic manipulators with Flexible links*, Ph.D. Thesis, Purdue University, Dec. 1984.

5. Petroka, R. P., *Computer Simulation And Experimental Validation Of A Dynamic Model (Equivalent Rigid Link System) On A Single -Link Flexible Manipulator*, M.S. Thesis, Naval Postgraduate School, Monterey, CA, Oct. 1982.

6. Gannon, K. P., *Modeling And Experimental Validation Of A Single - Link Flexible Manipulator*, M.S. Thesis, Naval Postgraduate School, Monterey, CA, Dec. 1986.

7. Cannon, R. H. and Schmitz, E., "Initial Experiments on the End Point Control of a Flexible One- Link Robot Flexible," The International Journal for Robotics Research, Vol. 3, No. 3, pp. 62-75, Fall, 1984.

8. MOOG INC. MOOG Series 760 Two-Stage Flow Control Servovalve Catalog, MOOG INC., EAST Aurora, N.Y.

9. Merrit, H. E., " Hydraulic Controls Systems", Wiley, N.Y., 1967.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Department Chairman, Code 69 Dept. of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Prof. L. W. Chang, Code 69Ck Dept. of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	6
5.	Dr. Kil Soo Kim, Code 69Ki Dept. of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Park,Ki Soon Busong Dong 534, I Ri City, Jon Ra Buk Do, Republic of Korea	7
7.	Air Force Central Library Sindaeban Dong, Kwanak Gu, Seoul,Republic of Korea	1
8.	Library of Air Force Academy Chongwon Gun, Chung Cheong Buk Do, Republic of Korea	1
9.	Department of Mechanical Engineering Air Force Academy Chongwon Gun, Chung Cheong Buk Do, Republic of Korea	3
10.	Prof. David Smith, Code 69Sm Dept. of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	1

11. Chief of Naval Operations
    OP-02
    Navy Department
    Washington, D.C. 20350-2000   1

12. Chief of Naval Operations
    OP-03
    Navy Department
    Washington, D.C. 20350-2000   1

13. Chief of Naval Operations
    OP-05
    Navy Department
    Washington, D.C. 20350-2000   1

14. Dr. Alan L. Meyrowitz, Code 433
    ONR
    800 N. Quincy St.
    Arlington, VA 22217   1